SOFE 3490
Software Project Management
Laboratory 3

Team

100617713	Samantha Husack
100620049	Alexander Hurst
100618756	Darren Chan

Project Title: Music Hub

Task 1: Estimated Effort

1.1 Function Point

End User Requirement	Type	Function points (complexity)
User can listen to music	Output	5
Artist can upload music	Input	5
User can specify what song they would like to listen too	Input	1
Artists songs are stored in a database	Internal files	2
Platform doesn't allow piracy	Outputs	3

1.2 COCOMO (Constructive Cost) Model

To define an estimated effort using the COCOMO model, we must first identify the type of system our software project is. We can define it as Organic because although it is a fairly complex project, the project attributes can be sourced in solved problems from the previous applications and the team members will only need nominal experience regarding the problem statements.

1.2.1 Basic Model

$$E = a(KLOC)^b$$

The constants a and b in this formula are 2.4 and 1.05 respectively when assuming the project type to be Semi-Detached.

If the KLOC (Kilo-Lines of code) has been agreed upon to be approximately 5000 lines of source code, the formula will appear as the following:

$$E = 2.4(5)^{1.05} = 13.0056 \text{ Person-Months}$$

Therefore, using the basic model, we can assume an approximate 13 months, roughly a year to complete this project.

1.2.2 Intermediate Model

$E = a(KLOC)^b * EAF$

The effort adjustment factor (EAF) is calculated using 15 values, derived from the 15 cost driver ratings proposed with this model. Each cost driver is given the rating of very low, low, nominal, high or very high. Each cost driver value is then multiplied together to calculate the total effort adjustment factor.

The below table demonstrates the cost driver values for each attribute.

Cost Driver	**Rating**	**Value**
Product Attributes		
Required Software Reliability	nominal	1.0
Size of Application Database	very high	1.16
Complexity of The Product	nominal	1.0
Hardware Attributes		
Runtime Performance Constraints	high	1.11
Memory Constraints	high	1.06
Volatility of the virtual machine environment	low	0.87
Required turnabout time	nominal	1.0
Personal Attributes		
Analyst capability	nominal	1.0
Application experience	high	0.82
Software engineer capability	nominal	1.0
Virtual Machine experience	high	0.90
Programming Language experience	nominal	1.0
Project Attributes		
Application of software engineering methods	nominal	1.0
Use of software tools	low	1.10
Required development schedule	high	1.04
EAF	**= 1.002509443**	

Taking the KLOC to be 5000, then the formula will appear as the following:

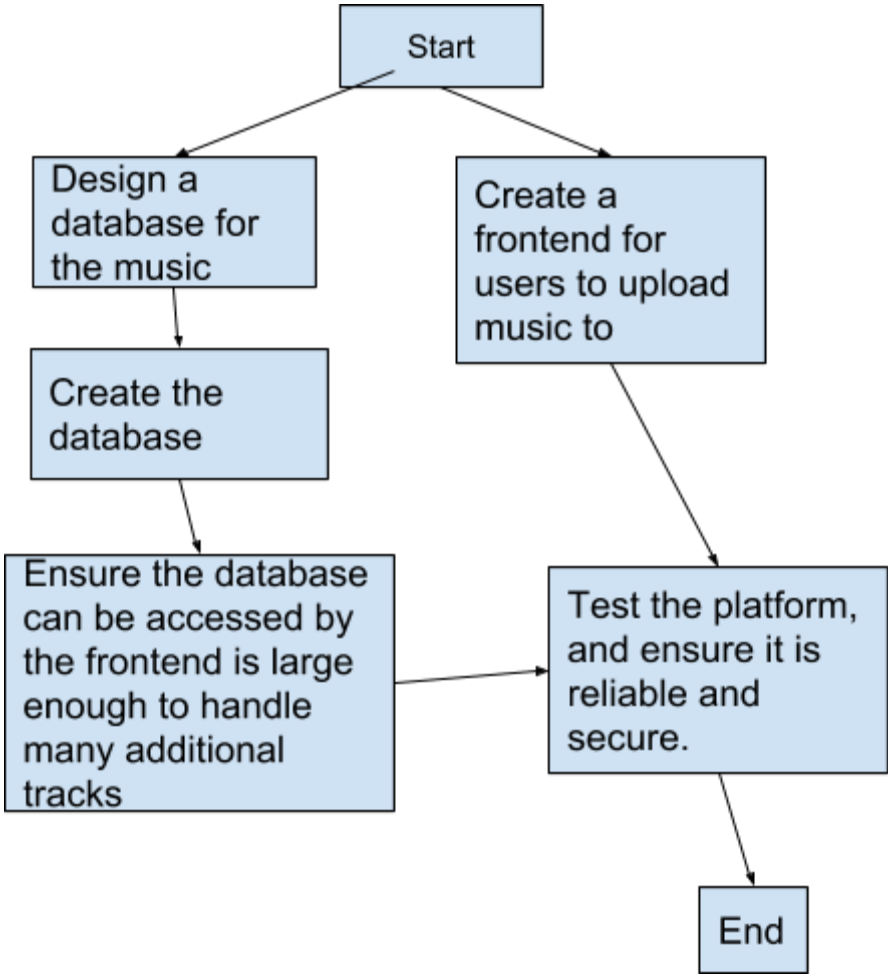
$E = 2.4(5)^{1.05} * 1.002509443 = 13.038217404 \approx 13.04$ Person-Months

Therefore, using the intermediate COCOMO Model, we can estimate an approximate 13 months for project development.

Taking into account both COCOMO models, the basic model as well as the intermediate model both estimate a project estimated effort of approximately 13 Person-Months.

Task 2: Project Activity Diagram

Here is our Project Activity Diagram



Task 3: Risks

There are multiple risks associated with the development of the project. Outlined below are descriptions of the kind of risks that the project could face and the countermeasures that will be placed:

1. As this project is a project with an estimated development time of 13 months, employee productivity could diminish
    a. Countermeasures would include a strict recruitment process as well as a rigorous training program before the beginning of the project.

2. Project tardiness, where elements might not be delivered on time, or certain requirements may take more work than planned. Scheduling issues in terms of development time estimations.
    a. Countermeasures would include a "cushion" period between product deadlines to provide a few days of grace in the case that a developer cannot meet a deadline.
    b. Monitor previous and existing projects and apply lessons learned.
    c. Provide estimation data including developers experience, project architecture, requirements, development flexibility, team cohesion, programmer capabilities, personal experience, project difficulty, constraints, …
    d. Furthermore, the possibility of overtime work or contract work will be taken into account in the cost analysis when project deliverables are to be larger than expected in the analysis.

3. Growth in requirements: as the project progresses, issues not identified at previous stages are identified and could create an issue with deadlines
    a. Countermeasures would include:
        i. Having extra time built into the schedule to handle any unforeseen circumstances
        ii. Testing the code every step of the development process to identify and fix bugs early in the development cycle.

4. A sudden employee turnover could happen if there are staff issues. Inso, a new member may be added to the team or a member may leave the team and this could cause a delay, or even derail certain aspects of the project.
    a. Countermeasures would include:
        i. Proper documentation of all developers work
        ii. Team meetings where all developers share progress reports
        iii. Proper resource collaboration
        iv. Proper knowledge sharing between team members
        v. Team collaboration on all development work

5. Procedural risks: day to day activities, holidays, unplanned days where the developers cannot get to the office to work, unplanned vacations

    a. Prepare in advance an employee attendance plan. Ensure that the project development takes into account the possibility of unplanned weather activity. (i.e. if the project will overlap with the winter months in Canada, take into account that there may be days that the developer team cannot commute to the office, or in the spring severe rain showers could delay team arrival to the office, etc.): countermeasures for these scenarios could include mobile workstations that allow developers to work from various locations, included delay approximations from historical data to be included in the time cost analysis.

6. Technical delays such as the need to compensate for an overdraft on the budget or time-schedule.
    a. Countermeasures would include a reduced functionality version of the product.
    b. A later release with an updated version, patches, updates might be planned to allow the product at reduced functionality to be released.

7. Unavoidable risks such as policies, obsolescence of software, hardware delivery delays, off the shelf software product that does not function as desired, etc.
    a. Countermeasures would include:
        i. Ensuring that everything we work with does not violate any rules or regulations that may be in place
        ii. Ensure that all the software we work with is up to date
        iii. Keep backups in case of any failures that cause us to lose data
        iv. Allow for ample time for hardware delivery, in the case that the shipment is delayed or otherwise late