A) Estimation by Proxy: Cost and Duration

UCP – Use Case Points

UCP = (UUCW + UAW) x TCF x ECF

UUCW: Unadjusted Use Case Weight

Here we have mapped the low, medium and high complexity use cases from use case survey in Appendix A to simple, average and complex use case point classification respectively.

Our use case model has 9 simple use cases, 6 average use cases and 15 complex use cases.

Use Case Category	Weight
Simple	5
Average	10
Complex	15

UUCW = (Total No. Simple Use Cases x 5) + (Total No. Average Use Case x 10) + (Total No. Complex Use Cases x 15)

= (9 x 5) + (6 x 10) + (2 x 15)

= 45 + 60 + 30

UUCW = 135

UUAW: Unadjusted Actor Weight

Similarly, we will calculate unadjusted actor weight for our use case model.

Assumptions:

Database is hosted on same network as the application and the implementation is simple. It has 5 main tables which are normalized to remove redundancy and improve integrity. It builds upon the proof of concept which was used for 3 years.

We have a total of 5 actors: Partner, Treasurer, Trader, System, Admin

The actor all is not considered individually as its activity is encompassed within the use cases of all other actors.

ACTOR	WEIGHT
Simple	System
Average	
Complex	Admin, Partner, Treasurer, Trader

System is a simple actor because it represents another system that performs automatic computation using online services. All persons/humans interacting via GUI are considered as complex actors.

UAW = (Total No. Simple Actors x 1) + (Total No. Average Actors x 2) + (Total No. Complex Actors x 3)

= (1 x 1) + (0 x 2) + (4 x 3)

= 1 + 0 + 12

UAW = 13

TCF: Technical Complexity Factor

Assumptions:

- The database used for the tool is in-memory cloud-based database which eliminates the need of a distributed system. The database and the application can be hosted on same network.
- The tool is available as a cloud solution and takes advantage of SAP HANA as a service to deploy the application anywhere which helps eliminate the complexities of on-premise installations.

The assigned value is the perceived impact of the technical complexity on the project.

Zero = no impact on the project

5 = strong impact on the project

Factor	Description	Weight	Assigned Value (Perceived Impact)	Weight x Assigned Value (Impact/ Factor)
T1	Distributed system	2.0	0	0
T2	Response time/performance objectives	1.0	2	2
T3	End-user efficiency	1.0	3	3
T4	Internal processing complexity	1.0	4	4
T5	Code reusability	1.0	2	2
T6	Easy to install	0.5	0	0
T7	Easy to use	0.5	2	1
T8	Portability to other platforms	2.0	0	0
T9	System maintenance	1.0	3	3
T10	Concurrent/parallel processing	1.0	4	4
T11	Security features	1.0	5	5

T12	Access for third parties	1.0	2	2
T13	End user training	1.0	4	4
Total (TF):				**30**

Compute the TCF using the formula: TCF = 0.6 + 0.01 × Σ Wi * Fi

TCF = 0.6 + (TF/100)

TCF = 0.6 + (30/100)

TCF = 0.9

ECF: Environmental Complexity Factor

Assumptions:

- The development team has been working on this tool since its inception and have done other projects in the past which makes them familiar with the process.
- The team has good technical knowledge, but they lack domain knowledge about stocks/bonds etc. for which they have collaborated with financial analysts to bridge the domain gap.
- Communication and coordination is a major issue as all the members are working part time. Team could experience low motivation at times so all E5 – E7 have a perceived high impact.
- Development team is building the tool using language/framework in which all the team is comfortable and has prior experience. So E1 – E3 have a low-medium perceived impact.

The weights for below calculation are for a dedicated and competent team.

Zero = no impact on the project

5 = strong impact on the project

Factor	Description	Weight	Assigned Value (Perceived Impact)	Weight x Assigned Value (Impact/Factor)
E1	Familiarity with development process used	1.5	2	3
E2	Application experience	0.5	3	1.5

E3	Object-oriented experience of team	1.0	2	2
E4	Lead analyst capability	0.5	3	1.5
E5	Motivation of the team	1.0	4	4
E6	Stability of requirements	2.0	5	10
E7	Part-time staff	-1.0	4	-4
E8	Difficult programming language	-1.0	1	-1
Total (EF):				**17**

ECF = 1.4 + (-0.03 x EF)

ECF = 1.4 + (-0.03 * 17)

ECF= 1.4+ (-0.51)

ECF = 0.89

Use Case Points

UCP = (UUCW + UAW) x TCF x ECF

UCP = (135 + 13) x 0.9 x 0.89

UCP = 118.548

Estimated Effort

Productivity is the effort it takes to develop one use case point. 20 is recommended value for any first project.

Assumption:

- Although the team has been working together for quite a while, this is the first time the project manager has been charged with project estimation. Owing of this, a Productivity Factor of 24 person-hour per use-case point is used. (Higher than the recommended 20 to account for intangible activities and risks)

Estimated Effort = UCP x Hours/UCP

Estimated Effort = 118.548 x 24

Estimated Effort = 2845.152 Person Hours

Estimated Time

Assumption:

- On average, a month has 160 'work' hours at 40 hours a week. Since the team is working part time we can safely assume each worker puts in 80 hours/ month.

Estimated Time = Estimated Effort / (part-time Hours/Week)

Estimated Time = (2845.152 Hours) / (80 Hours/Month)

Estimated Time = 35.5644 person-months

Estimated Cost

Assumption:

- The $5,500/person-month is **loaded cost**

Estimated Cost = Estimated Time x (Cost/Person-Month)

Estimated Cost = **35.5644** x $5,500/Person-Month

Estimated Cost = $195,604.2

Conclusion

This project will take an estimated **35.5644 person-months** to complete. That means, with **team of 5**, it should take approximately **7.11 months to complete** and cost an estimated **$195,604.2**

The staffing for this project includes 5 part-time employees where they multitask as testers as well and one of the employees is a first-time project manager. The duration and cost can be reduced if 2-3 developers can work full time in developing version one of the projects.

B) Estimation by Analogy

a) Productivity (function points/person-day) depends upon the language used rather than any other parameters. Function point efforts have a direct correlation to the SLOC of language being used. Language specific FP/person day is calculated in using the Mark II Function Point formula:

Unadjusted FP = 0.58 x (#of inputs) + 1.66 x (#of DB entities) + 0.26 x (#of output)
FP /person day = FP/effort

b) Productivity is directly proportional to SLOC in any given language so the FP per person-day is calculated using formula:

Unadjusted FP = 0.58 x (#of inputs) + 1.66 x (#of DB entities) + 0.26 x (#of output)

Project	Language	Function Points / Person Day
1	A	9.913333333
2	A	9.942117647
3	B	7.011851852
4	B	7.015403727
5	C	11.80272727
6	C	11.97428571
7	B	6.991623377
8	C	12.0816129

Computing averages for A, B and C from above values gives us productivity for each language.

Language	Productivity
A	9.92772549
B	7.006292985
C	11.9528753

c) For language B, productivity is approximately 7 function points per person-day. We can calculate the approximate effort by dividing the approximate function points for projects X and Y by the productivity of language B.

Calculated Function point value for project X is 704.66 and for project y is 261.8. Plug these values into formula: Effort= Function-Points/ Productivity

Effort(X) = Function-Points(X) / Productivity (Language B)
Effort(X) = 704.66 / 7.006292985
Effort(X) = 100.58 Person-Days

Effort(Y) = Function-Points(Y) / Productivity (Language B)
Effort(Y) = 261.8 / 7.006292985
Effort(Y) = 37.36 Person-Days

d) Project X shares similar inputs, outputs, and entities to project 3, and differs only in slightly smaller inputs and outputs. Using above formulae, we can compute that project X will take approximately 97 person-days.

Project Y has similar inputs, outputs, and entities to project 1 but has slightly lower value for all of parameters. They have a common language (language B), so they will be very similar, with Project Y being taking slightly less effort. Project Y will take approximately 37 person-days.

e) Yes, we should change the language. Language B has the lowest productivity (function-points/person-day). Language C would be the best choice because it has highest productivity.