Homework 3

Team 9 - John McFarren, Erica O'Kelly, Devila Bakrania, Matthew Monaco

"I pledge my honor that I have abided by the Stevens Honor System." - JM, EO, DB, MM

Summary: This assignment calls for the team to analyze and estimate for a system designed for reserving theater tickets. The team was given high-level OO specifications with actors and use cases. First, the team was tasked with estimating the amount of unadjusted and adjusted use case points for the project, and after adjusting, the team decided that the project included about 73 use case points. Additionally, the team needed to estimate the amount of staff weeks of effort required to complete the project. Using the Schneider & Winters method of calculating staff hours and incorporating a typical part-time worker's hours, the team soon came to the conclusion that the project would likely require around 82 staff weeks of effort. The team also devised alternative strategies to decrease effort needed for the project, in case the boss requested a 20% decrease in effort. The team found estimating the importance of technical complexity and environmental factors to be challenging, since this was very subjective, and many different assumptions could be made based on the information given. The team found it to be eye-opening to discuss the final part of the assignment, and consider that the "best" or most complicated solution isn't necessarily always the chosen solution for various reasons, and that some components and features may be considered excessive depending on the basic goal of the project.

1. How many UUCPs do you have? How many Adjusted Use Case Points? Why?

Unadjusted Actor Weight

Type	Amount of Actors	Total Weights
Defined API (1)	1	1
Interface with TCP/PIP type protocol (2)	0	0
GUI or a web page type interface (3)	3	9
UAW Totals	**4**	**10**

Unadjusted Use Case Weights

Type	Amount of Use Cases	Total Weights

Simple (5) < 4 transactions	3	15
Average (10) 4-7 transactions	3	30
Complex (15) > 7 transactions	0	0
UUCW Totals	**6**	**45**

UUCP = 10 + 45 = 55
Total Unadjusted Use Case Points would be 55.

Technical Complexity Factors

TFactor	Description	Weight	Imp	W * I
1	Distributed System Required	2	2	4
2	Response Time	2	4	8
3	End-User Efficiency	1	5	5
4	Complex Internal Processing	1	1	1
5	Reusable Code	1	2	2
6	Installation Ease	0.5	5	2.5
7	Usability	0.5	5	2.5
8	Cross-Platform	2	5	10
9	Easy to Change	1	3	3
10	Concurrent	1	4	4
11	Custom Security	1	5	5
12	Dependence on 3rd Party	1	5	5
13	User Training	1	1	1
		TOTAL		**53**

We arrived at 53 because the team for this project is looking to develop an application for reserving tickets at theaters. Since the customer will be reserving the tickets and the admins will be editing the events we felt the interfaces needed to be easy to understand to allow for better customer experience and efficiency for the admin.

TF = .6 +.01 * 53
Technical factor is 1.13

Environmental Factors

EFactor	Description	Weight	Imp	W * I
1	Familiar w/ Project	1.5	3	4.5
2	Application Experience	0.5	2	1
3	OO Experience	1	2	2
4	Lead Analyst Capability	0.5	2	1
5	Motivation	1	1	1
6	Stable Requirements	2	4	8
7	Part-time Staff	-1	5	-5
8	Difficult Programming Language	-1	5	-5
		TOTAL		**7.5**

We arrived at 7.5 because the team for this project is made up completely of part time workers using a new and tricky programming language. Since they are part time, we concluded their motivation would be low, and combined with the new language, their experience with application, OO, and lead analyst capability would also be low.

EF = 1.4 + (-.03 * 7.5) = 1.175

AUCP = UUCP * TF * EF

UCP = AUCP = 55 * 1.13 * 1.175 = 73.03

2. How many staff weeks of effort do you estimate?

- PF = 36 hours/UCP, using the Schneider & Winters method of calculating staff hours
- 32 real hours per week, which is typical of a part-time worker.
- Effort = 36 * 73.03 / 32 = 82.16 weeks = about 82 weeks

3. Your boss listens nicely to your estimates, and tells you they are 20% too large. How do you cut the effort by 20%? What can you do about it?

If the boss were to tell the team that the estimates were 20% too large, effort could be cut by 20% by essentially combining events into more inclusive events. For example, combining “make”, “cancel” and “lookup” reservation into a single, simpler event could reduce effort. Also, surveying the part-time team to see which software languages they’re most comfortable with and going with a language that the team is familiar with could reduce effort significantly. Giving the staff some input on this could increase motivation, decrease programming difficulty, increase experience, and greatly affect our environmental factors.