SOFE 3490 Lab 3:
Software Project Management

Wednesday, March 10th, 2020

Meghana Govind	100620126
Sheila Kuria	100618783
Tasfia Alam	100584647

Estimated Effort:

Inputs	Complexity
Customers information	low
Number of customers on one table	low
Table number	low
Lunch or dinner hour	low
Number of items	medium
Number of each item ordered	medium
Payment method(cashless restaurant)	high

Outputs	Complexity
Total Price of Order	medium
Number of Items purchased	low
Price of each item	low
Customer information on receipt	low
Order (going to kitchen)	high
Restaurant Information	low
Customer Loyalty Points	low

Queries	Complexity
Menu Items	medium
Customer Information (Loyalty Program)	low

External Files	Complexity
Menu Items	low
Customer Information	low
Stock (item availability)	medium

Internal Files	Complexity
Storing Order information for Table Number	medium
Storing & Updating Customer Loyalty Points	medium

Function Point Estimation:

Description	Complexity			Total
	Simple	Average	Complex	
Inputs	4 x 3	2 x 4	1 x 6	26
Outputs	5 x 4	1 x 5	1 x 7	32
Queries	1 x 3	1 x 4	_ x 6	7
External Files	2 x 5	1 x 7	_ x 10	17
Internal Files	_ x 7	2 x 10	_ x 15	20
			Total Function Points (FP)	**102**

Adjusted Function Point Estimation:

Environmental Factor	Rating
Data Communications	3
Distributed Computing	4
Performance Requirements	4
Constrained Configuration	3
Transaction Rate	5
Online Inquiry and/or Entry	1
End-user Efficiency	4
Online Update	1
Complex Processing	4
Reusability	3
Ease of Conversion/Install	2
Ease of Operation	2
Used at Multiple-Sites	2
Potential for Future Change	2
Total Environmental Factor (N):	**40**
Complexity Adjustment Factor(CAF): CAF = 0.65 + (0.01 x N)	**1.05**
Adjusted Function Points(AFP): AFP = CAF x FP	**107.1**

We will be using Java to code this restaurant ordering application. Therefore, the SLOC per function point for Java is 53. [1]

LOC = AFP x 53

LOC = 107.1 x 53

LOC = 5 676.3

KLOC ~ 5.67 thousand lines of code in Java

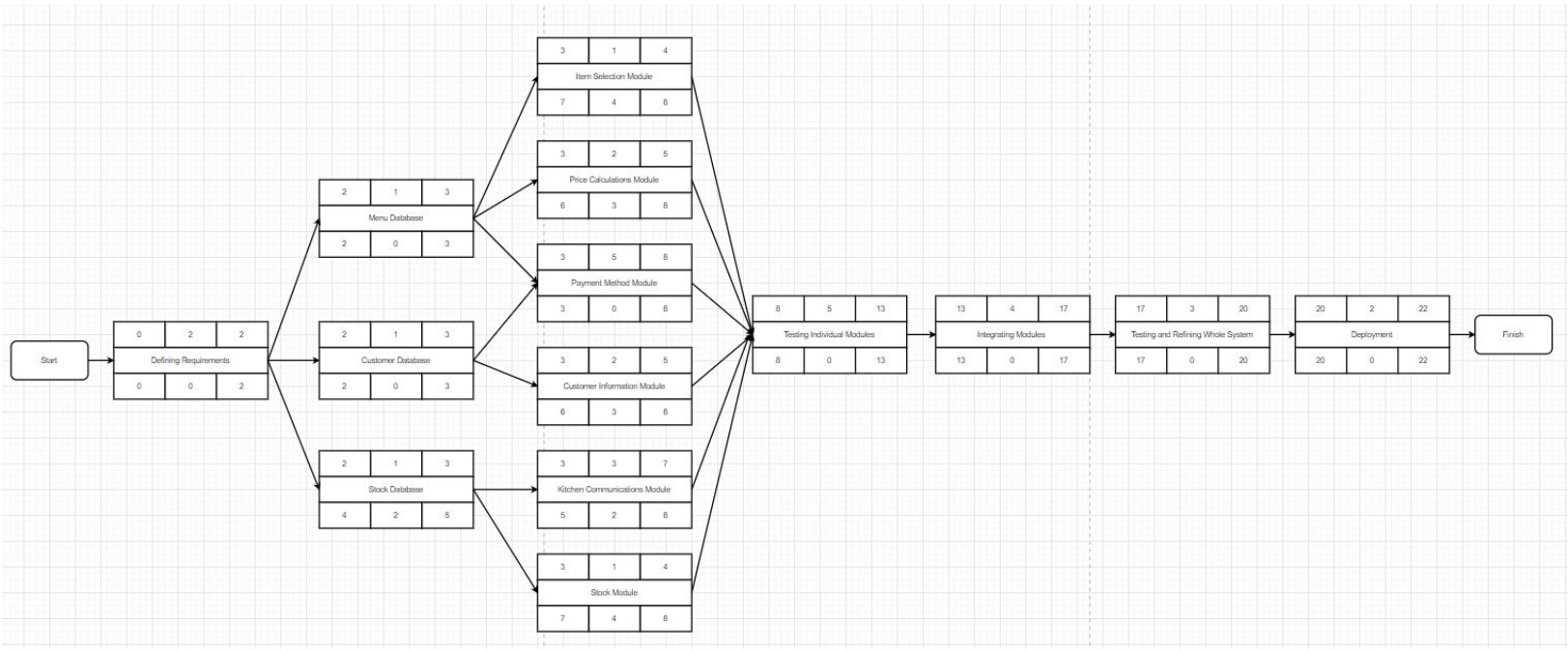
COCOMO:

Many restaurant ordering applications have been made before and there are many references online. This ordering application has straight-forward requirements and is a smaller size. We are a team of 3 with high team cohesion, however we are personally unfamiliar with this project. We are still learners and do not have sufficient experience creating such projects in Java. Therefore, we estimate this project to be a semi-detached project.

Semi-Detached: a = 3.0 b = 1.12 [2]
Effort = $a\ (KLOC)^b$
Effort = $3.0\ (5.676)^{1.12}$
Effort = 20.97 person-months

Activity Planning:

Everything is measured in number of days. Starting at day 0. First phase is defining and refining the requirements of the system. Second phase is creating the necessary databases. The third phase is coding the modules for the system. Next is testing each individual module, then integrating all the modules together to create the whole system. Then testing and refining the whole system. Finally, deployment.



Risk Management:

Risk	Description	Countermeasure
Late changes to requirements	New ideas and requirements might come during the development of the project. If using an agile process where the customers are involved, the customers might want to change/modify the existing requirements	Change is welcomed, especially when it improves the project, however welcoming new ideas should constantly be reviewed thus there is no delay to the project. We should also implement the requirements that will take the most time so if there is a delay there is something to submit. .
Staff Limitations	Staff may not be as productive as expected, due to limitations in experience and this may slow down the project	- Make sure high quality staff are hired - Make sure that all employees have good training - Make sure that teamwork is emphasized throughout process
Impractical Estimate	Over estimating how much time or resources a certain task will take to implement.	Try to finish the project a few weeks before the deadline so that if any issues come up there is a reasonable time frame to correct any errors and missing requirements before the real deadline.
Security Issues	We are dealing with customer's private information there are chances for data leaks and unauthorized access	- Ensure we have proper protection/security against ill-intent

References:

[1] Quantitative Software Management. "Function Point Languages Table," *Quantitative Software Management Inc.* 2017. [Online]. Available:

https://www.qsm.com/resources/function-point-languages-table [Accessed: March 10, 2020].

[2] A. Abdalbari. SOFE 3490. Class Lecture, Topic: “Chapter 5: Software Effort Estimation.” Department of Electrical, Computer, and Software Engineering, Ontario Tech University, Oshawa, ON. Feb 3, 2020.