Final Project of SSW-533A

TEAM PROJECT REPORT

The Practice of Cost Estimation and Metrics in Software Engineering Based on Specified Topic

TEAM No. 3

Dongjie Zou

Shengping Xu

Yujun Kong

Zening Fang

Abstract

1.01 Summary

Frankly speaking, at the moment of just getting the requirement of contriving this final project and after the formally having formed our team, we totally have no idea to conduct a qualified technical team project report. We once briefly held a short meeting after the class in the library, yet still, we cannot find a proper technical topic to match the following works in sizing and estimating for a software development life cycle activity in simulation mode.

Many once raised the thought of software system topics or categories in the selection for resulting in a technical report are inappropriate and overthinking, which seems applicably unrealistic or scientifically complicated. However, with the deeper thinking progress between the teammates' brainstorming, a good idea emerged. We asked ourselves, why can't we combine the real happenings that occur in the real world to apply to this assignment for the final project?

Therefore, we found a quite suitable application and technology topic for the software types of selection for the works of effort estimation and development measurement. According to the current situation of insane lockdown in Shanghai City China where the people are suffering from the unwise and stubborn COVID policy, we saw the people have to do self-rescue in the supplement of foods, medicines, and groceries via the approach of group-purchasing. This miserable story did inspire us to like to develop a software that can champion the livelihood emergencies of the people who are getting affliction in Shanghai City.

So, we elicited the core requirements and conducted a series of feasible processes of achieving matched metrics, measurements, and predictions for software development. Those works consist of stakeholders identification, software feature and component specification, function points conduction, use case points generation, LOC of the whole project estimated calculation, and defect density prediction. Now, we begin to provide these elucidations in the following chapters, but, before that, let's first check the table of the contents of this writing.

1.02 Table of Contents

Introduction of Background, Stakeholders, and Software Look and Feel

2.01 Background Story

Whatever and however the channels of the Chinese government's official press explained, or some international media described, in reality, the people in Shanghai City are indeed suffering the dilemma

of the highly pressured lockdown from the so-called "Dynamic Zeroing" policy of fighting the COVID-19 virus pandemic. Because of the extremely restricted stipulation of locking down by the government, the people are forced to be required staying their homes without any possibility of walking out, besides a very few amount of the people who are living in some luxury neighborhoods with especially advanced living supplements, most people have to be nowadays struggling for their living conditions every day. Even though realistically, the living stuff on the market and society, in general, shall be ample, the reality sadly shows that the shortage of the supplement still exists, rendering the annoying, complaints, and desperate from the people.

The people have to count on some technical approaches to keep their living conditions under control by purchasing the goods of food, medicine, and grocery online. Nevertheless, the traditional buying way in the mode of point-to-point, which means each family just purchases the living necessities respectively without any relationships between the near-door neighbors, in the same building or neighborhoods community block, has the limitation of the burdening capacity of the unified orders management, rational logistics, and goods dispatching under the control by the local pandemic-handling personnel. The living routine of the people in communities in Shanghai City is no longer like the formerly normal days, has been totally changed, every family cannot freely take their bought living stuff from the door of the apartment via the delivery from the online stores' logistic service guys.

The people started to unitedly purchase their living goods and materials in the group mode based on their geographically living scales such as a townhouse, an apartment building, and a community area or street block. Nonetheless, the relevant internet applications for group purchasing are very lacking proper functionalities design and inadequate comprehensive management dealt with by the local neighborhood and its pandemic officials. Thus, we like to develop a good enough software that can well solve the problems of purchasing the living goods in group mode and help the Shanghai people presently living in such pitiable circumstances by fighting against the pandemic.

2.02 Software Feature and Functionality

Based upon the above rough understanding of the background story, the features and functionalities of this software system now could be derived from the requirements' specification via the identification of the stakeholders. They embody a set of the following roles with their respective business requirements in the software system:

Table of Stakeholders and Related Requirements

Roles	Requirements

End-Users	
Each Family	Needs to buy their living goods such as daily foods, medicines, and groceries.
Managing Users	
Apartment Principal	Needs to manage every family's order and coordinate the goods dispatch and allocation for the residents of the current apartment building.
Community Block Principal	Needs to manage each apartment building's goods allocation and delivery, reviewing the transaction or payment status and their details for the whole community block.
Community Official (The Director of Local Resident Committee)	Needs to manage all community blocks in the living supplement, payment and transaction details, logistic processing, poor families' special supplements, etc. Needs to report the daily logs to the higher management.
District Official	Needs to review the present district routines, lead the policy implementation, and report to the higher management.
Supplying Users	
Living Goods Supplier	Needs to list the daily supplement details and provide these to the business assigned community block principals, organizing the service of logistics.
System Owners	
Development Teams	Need to develop, test, release, deploy, fix, and update the software system.
System Admins	Need to maintain the software system.

Possibly Non-Functional Features of the Software System	
Features	Involved Roles
Viewing Goods	End-Users, Managing Users
Listing Goods	Managing Users
Searching Goods	End-Users, Managing Users
Editing Goods	Managing Users, Supplying Users
Ordering Goods	End-Users
Paying Orders	End-Users
Listing Orders	Managing Users

Searching Orders	Managing Users
Editing Orders	End-Users, Managing Users
Editing Own Family Info	End-Users
Listing Families	Managing Users
Editing Families Info	End-Users
Listing the Current Apartment Transactions	Managing Users – Apartment Principals
Managing Bought Goods Allocation	Managing Users – Apartment Principals
Listing the Current Community Block's All Apartments Details in Goods, Orders, Payment, Dispatch, Delivery, Anti-COVID Treatment for the Bought Goods	Managing Users – Community Block Principals
Listing the All-Managed Community Blocks' Information in Living Goods Supplement	Managing Users – Community Officials
Listing the Current City-District's Community Blocks' Information in Living Goods Supplement	Managing Users – District Officials
Listing the Living Goods Supplement Targeted Community Blocks'	Supplying Users – Living Goods Suppliers (Business Entities of Online Stores)

Information in Transactions, Logistics	

Generally speaking, on the client-side, this software system shall be acting like a web-based and WeChat embedded application running on the lockdown people's smart devices such as iPhones or Pads, Android Phones or Pads, Laptops or Desktops. On the server side, this application should be running on the cloud service and holds its independent database that also can be linked to some government management platforms to provide the statistics ability for helping the relevant authorities to better the residency service on the scales of the districts or entire Shanghai City.

Proposed Metrics

3.01 Reasoning and Mattering of Applicable Metrics

According to the discussed conclusions from two times our meetings as a team and the experiences of our past assignments homework, we determined to use the metrics of the Function Points method and the Use Case Points method. We have the consent that we all believe these two metrics in sizing the final effort and duration of the development are quite reasonable, reliable, and pretty useful and practical. So, for concreting the persuasiveness of the project based on this software system development that probably would be implemented shortly, we like to apply these two methods to estimate thence develop this software system and now let's see how the eventual outcomes in software sizing measurement could be at the bellowing.

3.02 Metric of Function Points Measurement

Count the Value of Function Points	
Identify Basic Function Points	
#	Description
Internal and External Inputs	
01	Families' Citizens and Residence Information (S)
02	Families' Selected Goods Cart Information (A)
03	Families' Order Payment Transaction Information (S)
04	Families' Orders Information of Bought Living Goods (C)

05	Apartments' Citizens and Residence Information (S)
06	Community Blocks' Citizens and Residence Information (S)
07	Apartments, Community Blocks Citizens and Residence Information (S)
08	Communities, Districts Officials Citizens and Residence Information (S)
09	Government Arranged Anti-Covid Personnel's Citizens and Residence Information (S)
10	Living Goods Supplement Providers' Business and Logistics Information (S)
11	Living Goods Supplement Providers' Foods, Medicines, Groceries Offering Information (C)
12	API from WeChat and Alipay Platforms (A)
13	API from Banking Payment Systems (A)
14	Software System's Own Admins Inputs (C)
External Outputs	
01	Data through API to the city government-owned statistics platform in all residents living conditions goods and materials management under the current situation of anti-pandemic for sending the statistical data generated by this software. (A)
02	Data through API to the city government-owned statistics platform in all residents living conditions goods and materials financial aid under the current situation of anti-pandemic for sending the statistical data generated by this software. (A)
03	Data through API to WeChat which most residents are using. (A)
04	Data through API to Alipay which most residents are using. (A)
05	Data through API to the respective banking payment systems for a web-based version of the software. (C)
06	Data through API to the respectively each living goods supplement business providers for collaborating the orders and logistic information. (S)
Logical Internal files	
01	End-User (Each Family User) Interfaces (A)
02	Apartment Building Principal Interfaces (A)
03	Community Block Principal Interfaces (C)
04	Community Official (The Director of Local Resident Committee) Interfaces (C)
05	District Official Interfaces (A)
06	Statistical Data Items Listing and Visualization Interface for Shanghai City-Level Authorities' Officials Review (C)
07	General Database (C)
08	Backup Databases (S)
External Interface files	
01	API of Banking Payment Systems (No Need to Implement)
02	API of WeChat and Alipay Platform (No Need to Implement)
03	API of Government Review of Statistics (Highly Possibly Need to Implement) (S)

04	API of Directly Manageable Logistics Service Providers Who Have Own Info Systems (Highly Possibly Need to Implement) (S)
External Inquiries	
01	Inquiry from Living Goods Supplement Enterprise that Has Some Platform (S)
02	Inquiry from Directly Manageable Logistics Service Providers Who Have Info Systems (S)
03	Inquiry from Government Residence or Anti-Pandemic Platform for Statistical Strategies (S)

Weight Complexity Ratings:

Attribute	Simple	Average	Complex	Quantity	Point
Inputs (Simple)	3	4	6	8	24
Inputs (Average)	3	4	6	3	12
Inputs (Complex)	3	4	6	3	18
Outputs (Simple)	4	5	7	1	4
Outputs (Average)	4	5	7	4	20
Outputs (Complex)	4	5	7	1	7
Data Files (Simple)	7	10	15	1	7
Data Files (Complex)	7	10	15	1	15
Interface Files (Average)	5	7	10	3	21
Interface Files (Complex)	5	7	10	3	30
Inquiries (Simple)	3	4	6	3	9
				UFPs	**167**

Calibrate with VAF-Calculating Based on GSCs

General System Characteristics

FP Complexity Adjustment Weights		Impact Weight						
#	Category:	No	Minor	Mod	Avg	Sig	Strong	Value
		0	1	2	3	4	5	
01	Data Communications					4		4

02	Distributed Data/Processing						5	5
03	Performance Objectives					4		4
04	Heavily Used Configuration						5	5
05	Transaction Rate						5	5
06	On-Line Data Entry						5	5
07	End-User Efficiency					4		4
08	On-Line Update					4		4
09	Complex Processing				3			3
10	Reusability			2				2
11	Conversion/Installation Ease					4		4
12	Operational Ease				3			3
13	Multiple Site Use				3			3
14	Facilitate Change					4		4
							VAF	**55**
AFPs = UFPs * (0.65 + 0.01 * VAF) = 167 * (0.65 + 0.01 * 55) = **200.40** Function Points								

Now, we just temporarily put the counted value of Function Points aside for the next step of estimating the Effort of the Staff Months and the LOC of the whole software system's implementation works.

3.03 Metric of Use Case Points Measurement

Calculation of UUCP				
Calculate UAWs				
Actor:	Complexity:			Weight:
	Humans = 3	Protocols = 2	Defined API = 1	
01	End-Users			3
02	Managing Users			3
03	Supplying Users			3
04	System Owners			3
05		TCP/IP for WeChat		2
06		TCP/IP for Alipay		2
07		HTTP for Web App		2
08		Stored Procedures for the System's Database		2
09		Government Data VPN		2
10			API to Government's Residents' Living	1

			Conditions Supplement Management Platform	
11			API to Government's Residents' Financial Aid Management Platform	1
12			API to WeChat Platform	1
13			API to Alipay Platform	1
14			API to Payment Banking Systems	1
15			API to Living Goods Supplement Business Enterprises	1
16			API to Government's Residents Situation Statistical Systems Management Review	1
17			API to Government's Directly Manageable Logistics Authorities	1
Total:(Sum)				**30**
UAWs = **30**				
Calculate UUCW				
Use Case:	Complexity:			Weight:
	<4 Transactions = 5	4-7 Transactions = 10	>7 Transactions = 15	
Family				
01	View 1 Transaction			5
02	Add 4 Transaction			5
03	Delete 3 Transaction			5
04	Seek 2 Transaction			5
05	Modify 4 Transaction			5
			Sub	**25**
Apartment Building				
06	View			5

	1 Transaction			
07	Add 4 Transaction			5
08	Delete 3 Transaction			5
09	Seek 2 Transaction			5
10	Modify 4 Transaction			5
			Sub	**25**
Community Block				
11	View 1 Transaction			5
12		Add 6 Transaction		10
13	Delete 4 Transaction			5
14	Seek 2 Transaction			5
15		Modify 6 Transaction		10
			Sub	**35**
District				
	View 1 Transaction			5
	Seek 2 Transaction			5
	Modify 4 Transaction			5
			Sub	**15**
Goods				
	View 1 Transaction			5
		Add 4 Transaction		10
	Delete 3 Transaction			5
	Seek			5

	2 Transaction			
		Modify 4 Transaction		10
			Sub	**35**
Order				
	View 1 Transaction			5
		Order 4 Transaction		10
	Seek 2 Transaction			5
		Modify 4 Transaction		10
			Sub	**30**
Payment				
	View 1 Transaction			5
		Pay 6 Transaction		10
	Seek 2 Transaction			5
		Modify 5 Transaction		10
			Sub	**30**
Allocation				
	View 1 Transaction			5
	Plan 4 Transaction			5
	Delete 3 Transaction			5
	Seek 2 Transaction			5
	Modify 4 Transaction			5
			Sub	**25**
Logistics				

	View 1 Transaction			5
	Add 4 Transaction			5
	Delete 3 Transaction			5
	Seek 2 Transaction			5
	Modify 4 Transaction			5
			Sub	**25**
Personnel				
	View 1 Transaction			5
	Add 4 Transaction			5
	Delete 3 Transaction			5
	Seek 2 Transaction			5
	Modify 4 Transaction			5
			Sub	**25**
Message				
	View 1 Transaction			5
	Edit 2 Transaction			5
	Send 1 Transaction			5
	Delete 2 Transaction			5
	Seek 3 Transaction			5
			Sub	**25**
Report				
	View 1 Transaction			5

	Edit 2 Transaction			5
	Send 1 Transaction			5
	Delete 2 Transaction			5
	Seek 3 Transaction			5
			Sub	**25**
Statistics				
	View 1 Transaction			5
		Configure 5 Transaction		10
	Export 3 Transaction			5
			Sub	**20**
System				
	View Log 1 Transaction			5
			Configurations > 10 Transaction	15
	Export Log 4 Transaction			5
			Sub	**25**
Total:(Sum)				**365**
UUCW = **365**				
UUCP = UAW + UUCW = 30 + 365 = 395				

Calculation of AUCP				
Calculate TF (Technical Complexity Factors)				
TC-Factor:	Description:	Weight:	Complexity:	W * C
01	Distributed System	2	2	4
02	Response or Performance Objectives	2	2	4
03	End-User Efficiency	1	2	2
04	Complex Internal Processing	2	2	4
05	Reusable Code	1	1	1

06	Easy to Install	0.5	1	0.5
07	Easy to Use	0.5	1	0.5
08	Portable	2	1	2
09	Easy to Change	2	2	2
10	Concurrent	2	2	4
11	Security Objectives	2	2	4
12	Provides access for 3rd Parties	3	2	6
13	Special User Training Facilities Required	1	1	1
Total:(Sum)				**35**
TCF = 0.6 + (0.01 * 35) = **0.95**				
Calculate EF (Environmental Factors)				
E-Factor:	Description:	Weight:	Value:	W * V
01	Familiarity With Project	3	2	6
02	Application Experience	2.5	2	5
03	OO Experience	2.5	2	5
04	Lead Analyst Capability	3	3	9
05	Motivation	2.5	2	5
06	Stable Requirements	3.5	2	7
07	Part-Time Workers	- 1.5	4	- 6
08	Difficult Programming Language	- 2.5	2	- 5
Total:(Sum)				**26**
EF = 1.4 - (0.03 * 26) = **0.62**				
AUCP = UUCP * TCF * EF = 395 * 0.95 * 0.62 = 232.655 ≈ 233				

Now, we also just temporarily put the counted value of Use Case Points aside for the next step of estimating the Effort of the Staff Months of the whole software system's implementation works.

3.04 Predicted Effort of Staff Months Based on the Above Two Metrics

Count the effort based on Function Points with CMMI Level Metric			
As our respectable Professor Dr. Ens advised, we use the method of CMMI Level to calculate the effort.			
CMMI Level Calculation			
SEI Maturity Level		Meaning	Function Points per Staff Month
1	Initial	Chaotic	3
2	Repeatable	Marginal	3.5
2.5	Our	Somewhat Adequate	4.75

3	Defined	Adequate	5
4	Managed	Excellent	7.5
5	Optimizing	State of the Art	9

We are a team that consists of 4 guys with different coding talents, so we now make a table evaluating each developer's programming craft. See the bellowing:

Name	Dongjie Zou	Shengping Xu	Yujun Kong	Zening Fang
Skillfulness Level	Adequate	Excellent	Chaotic	Marginal
Level Score	5	7.5	3	3.5

Our Capacity in Programming as a Team

(5 + 7.5 + 3 + 3.5) / 4 = 19 / 4 = **4.75**

We have known that the formerly calculated Adjusted Function Points value is 200.40

Therefore, Effort = AFPs / (FP per Staff Month) = 200.40 / 4.75 = **42.1895 ~ 42** Staff Months

Basically, we can say, as a development team, we four guys can accomplish the coding implementation during 42 / 4 ~ **10.5 Months**

Count the effort based on Function Points with CMMI-Like Level Metric

Moreover, we try to count out the Effort by the value of Use Case Points.

CMMI Level-Like Calculation

We try to use a CMMI Level-Like Pattern (Our Customized Pattern) to count out the Effort by different values listed in the below table, which is a very fun shot!

Customized Maturity Level		Meaning	Needed Hours per Use Case Point
1	Initial	Chaotic	30
2	Repeatable	Marginal	25
2.5	Our	Somewhat Adequate	22.5
3	Defined	Adequate	20
4	Managed	Excellent	15
5	Optimizing	State of the Art	10

We are a team that consists of 4 guys with different coding talents, so we now make a table evaluating each developer's programming craft. See the bellowing:

Name	Dongjie Zou	Shengping Xu	Yujun Kong	Zening Fang
Skillfulness Level	Adequate	Excellent	Chaotic	Marginal
Level Score	20	15	30	25

Our Capacity in Programming as a Team

(20 + 15 + 30 + 25) / 4 = 90 / 4 = **22.5**

We have known that the formerly calculated Adjusted Use Case Points value is 233
Therefore, Effort = AUCPs * Hours per AUCP = 233 * 22.5 = **5242.5** Staff Hours
According to the condition that we are graduate students as the kind of Semi-Part-Time Developer, we probably could work 30 hours per week, So:
5242.5 Staff Hours / 30 hours per week = 174.75 Weeks Thence, 174.75 / 4 weeks = **43.6875** Staff Months
Basically, we can say, as a development team, we four guys can accomplish the coding implementation during 43.6875 / 4 ~ **11 Months**
Brilliant! These two estimated values of basing on metrics of Function Points and Use Case Points are incredibly close!!! It looks that our estimation by using the methodologies of Function Points and Use Case Points is truly SUCCESSFUL! We did apply the appropriate and wise metrics to sizing this software system project!

3.05 Estimated LOC Calculation Based on the Metric of Function Points

Count the LOC of Software System's Coding Works				
According to QSM Function Point Programming Languages Table (Selected Language)				
Programming Language	Avg	Medium	Low	High
ASP	51	54	15	69
Brio	14	14	13	16
C++	50	53	25	80
Excel	209	191	131	315
FoxPro	36	35	34	38
Java	53	53	14	134
Perl	25	15	15	60
SQL	21	21	13	37
Visual Basic	42	44	20	60
According to the before calculations and evaluations of our coding capacity, our skillfulness in programming is a bit lower than the average craft level, so we use 70 LOC per Adjusted Function Point to count out the total LOC of this software system development project, Therefore:				
LOC = AFPs * (QSM-FPPLT{Java}) = 200.40 * 70 = **14,028.00** Lines of Codes				

3.06 Defect and Its Density Prediction

Till now, although we have not yet conducted the coding implementation and testing execution, still, we like to testify if we have learned knowledge well for this course SSW533. So, we too love to do the prediction for the defect density in the following way.

Estimate Defects										
Since our estimated development duration as a team working is 10.5 months, so:										
Given Data from Debugging Test Results										
Development Months	1	2	3	4	5	6	7	8	9	10
Defects Found	20	35	16	33	19	28	N/A	N/A	N/A	N/A
Problem Solving Mindset and Mind-Flow										
For predictably counting out the remained defects based on given conditions, there is a simple approach that can be used by a pattern of Rayleigh Distribution. It's based upon ~40% of the total defects that have appeared by Tm. Therefore, the calculation shall be like the bellowing.										
The Calculation for K (Totally Possible Defects)										
K = (20 + 35 + 16 + 33) * (100 / 40) = 104 * 2.5 = **260**										
So, each of 7th, 8th, 9th, 10th months shall on average be [260 - (20 + 35 + 16 + 33 + 19 + 28)] / 4 = **27.25**										

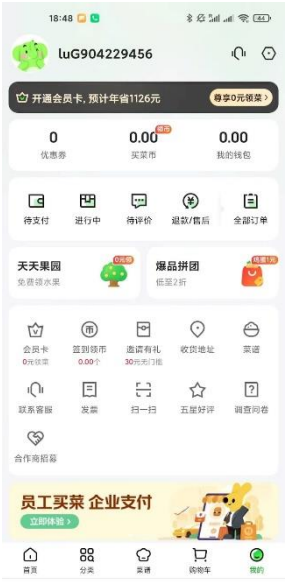
Results and Discussion

4.01 Result

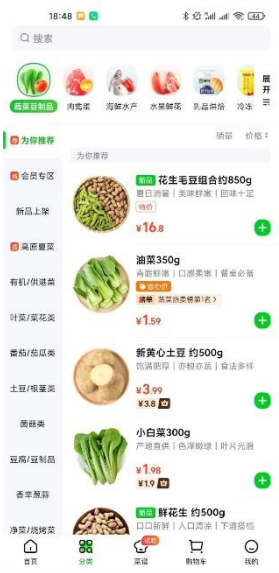
The current situation in Shanghai has exceeded expectations, with most residents locked down at home. We designed this software to meet the daily shopping needs of residents in quarantine.

The following are the demo displays:

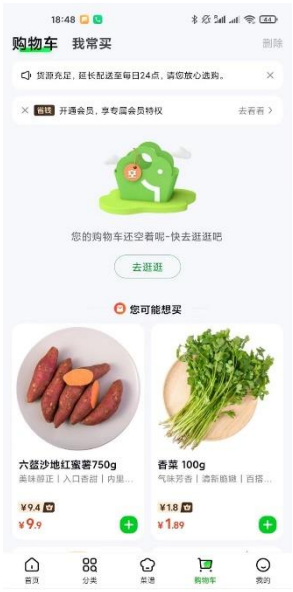
My Information Page:



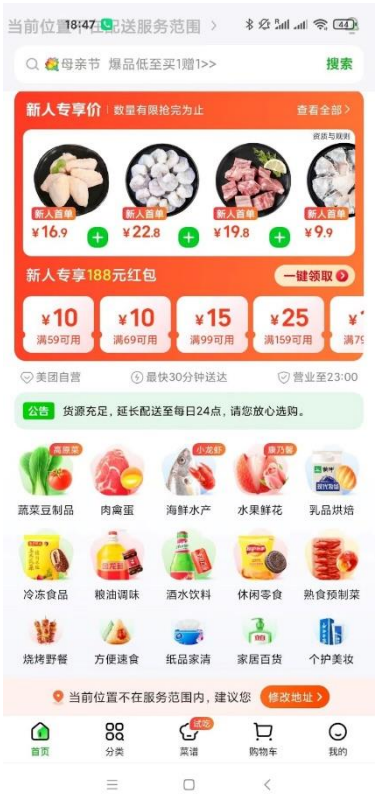
Category Page:



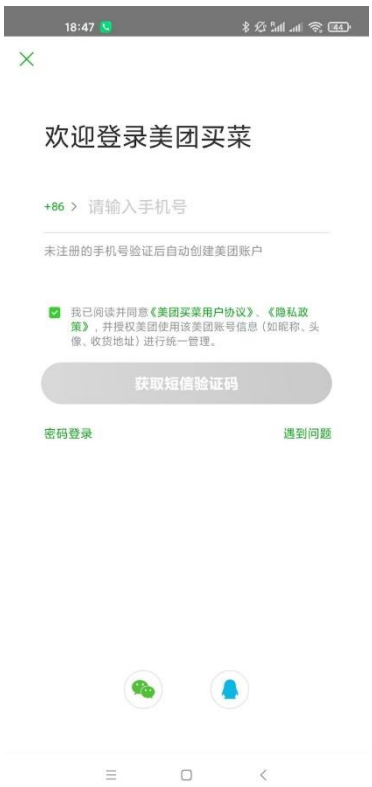
Shopping Page：



Main Menu:



Login Page:



For Login Page: We allow mobile phone and WeChat verification login.

Main Menu Page: You can browse popular products, click the page to browse product details.

Shopping Page: The status of the goods can be changed, and the purchased goods can be displayed in detail. The checkout button will jump to the checkout page.

My Information Page: You can view the status of the goods and modify the status of the order. You can also check whether the area is blocked, whether the logistics can be delivered, and the status of individuals and families.

4.02 Discussion

After reviewing and discussing the current estimation metric we used in this project, we have come up with the following good points and drawbacks in this project:

In this estimation project, we have used both the Use Case Point metric and Function Points metric. Using both metrics might cause more effort and time for our team, but it can make the estimation result more accurate.

In the LOC estimation part, we have assumed that we use Java as our programming language. However, in real production circumstances, we have to use different languages to implement different parts of the project. For example, we use JavaScript as the front end, Java as the back end, and MySQL as our database. This could cause changes in the gearing factor. Therefore, we could combine the gearing factor of different languages and tools to get a more accurate gearing factor for our project.

In the Defects Estimation part, we applied the Rayleigh Distribution model to estimate the arrival rate of defects in our project. However, we cannot make the estimation very accurate since we don’t have historical data from our team. We can only get an approximate result.

Limitations

5.01 Limitations of Used Metrics and Methodologies

It is difficult to estimate effort/size related to projects accurately, because at the planning phase of the

project there is uncertainty about the project scope, due to the rapid change in the requirements in agile software development. In our project, our target user's size may vary based on different environments. For example, some communities in relatively remote locations tend to have fewer residents, while in some residential intensive areas, one community may include over twenty housing estates, which may be occupied by thousands of families. As the number of users increases, it is hard to define the accurate database capacity and processing speed. Some systems hold huge volumes of data. FPA rarely considers data storage.

In addition, it does not consider the size of simulations, animations, and additional document effects. For the use case points estimation, we find that it may work well for creating a rough, initial estimate of the overall project size, they are much less useful in driving the iteration-to-iteration work of the team. For example, when we design the user interface for different function pages, we find some of the details are not reflected in the initial technical cases, and some estimations with big weight may have a small impact across the overall project. With that being said, we cannot fully provide all possible use cases, otherwise, there is no learning based on working software during this period.

Conclusion

6.01 Conclusion of Teamplay and Achievement

In the teamplay performance, we cooperated quite well. The morale was always kept on a very high level of the on-time meeting, rapid responses, quick critical thinking, closely collaborating, good looks and feel of the documentation publishment, and other qualified characteristics.

In the consideration of our achievement of this final project, we have the confidence to say that we literally conducted a fully qualified final project report even which is very close to the industrial standard. There are several shining parts of the accomplishment as the brilliant features we made.

1) We elicited a creative but still very empirically and realistically applicable topic in software development and its cost, effort, sizing, prediction, and estimation with proper metrics using. According to our outcomes for this final project, a new type of Group-Purchasing mode application we inspired seemly has a high possibility to be in the feasibility.

2) We applied the appropriate estimation and measurement methodologies such as “Function Points”, “Use Case Points”, “CMMI Level”, “SEI Maturity”, “Rayleigh Modeling”, etc.

Especially, two calculated outcomes of leveraging “Function Points” and “Use Case Points” metrics are wisely and incredibly close in quantitative concern. Moreover, our made identifications and specifications have been proved that our detailed and contextual calculations and prediction are also precise and persuasive.

What we have achieved represents enough expectation and outlook in executability and feasibility of this final project’s proposed software system development. We believe that, in near future, we will highly probably begin to implement this software system to help the people in Shanghai city who are still in the lockdown crisis of lacking living supplement goods.

References

7.01 References

1. Teaching Slices of Stevens Tech and Lecturing Contents from Professor Dr. Ens
2. The Textbook - Software Measurement and Estimation A Practical Approach by Linda M. Laird, M. Carol Brennan (z-lib.org)

Appendices

8.01 Appendices

None in this version of writing

VERSION: BETA - 0.95

APPRECIATIONS

At the end of this writing, we want to unfeignedly appreciate you can tolerate our somewhat verbose elucidation of this merely qualified article for understanding the knowledge of Software Cost Estimation and Metrics.

Additionally, we appreciate our Professor Dr. Richard Ens taught us so well to let us learn so many worthy practical techniques and theoretical notions in the lecturing content of the course SSW533.

Thank you for your time in reviewing this document!

TEAM No. 3

Dongjie Zou

Shengping Xu

Yujun Kong

Zening Fang

May 9, 2022