



INSTITUTE OF INFORMATION TECHNOLOGY
UNIVERSITY OF DHAKA



Software Metrics

Course : SE-611

Topic: Function Point Analysis of T-Commerce

Submitted to:

Dr. Emon Kumar Dey

Associate Professor

Institute of Information Technology

University of Dhaka

Submitted by:

Swadhin Pal - 1302

Amit Kumar Roy - 1314

Fareya Azam - 1331

Submission date: 28 January, 2025

Table of content:

Table of content:	2
1. Introduction to T-Commerce	2
1.1 Authentication	2
1.1.1 Login	3
1.1.2 Forget Password	3
1.2 Dashboard	3
1.3 Shopping System	4
1.4 Product Management	5
1.5 Notification Module	6
2. Identifying Function Points	7
3. Degree of Influence (DI)	11
4. Technical Complexity Factor(TCF)	12
5. Function Points	13
6. Estimate Lines of Code (LoC) from Function Points	13
7. Cost Estimation	17
References	19

1. Introduction to T-Commerce

T-Commerce is a Telegram bot-integrated e-commerce platform designed to provide a seamless and efficient shopping experience for both buyers and sellers. This innovative platform harnesses the user-friendly Telegram interface to revolutionize online commerce, offering robust features such as user authentication, product management, shopping, and notifications. T-Commerce combines the convenience of Telegram with powerful e-commerce functionalities, offering an intuitive, secure, and engaging shopping experience. It's a game-changer for both buyers and sellers in the evolving digital marketplace.

1.1 Authentication

To use this bot ,all users must register and login in their telegram account.

Admin Authentication

- Admins are pre-registered and do not need to sign up.

Customer Authentication via Telegram

- Customers initiate the bot using the "/start" command.
- Upon starting, customers are prompted with a "Share Your Phone Number" button.
- Customers provide their phone number to register.

Users are already authenticated by Telegram, ensuring secure entry into the system. By leveraging Telegram's built-in authentication, T-Commerce ensures a streamlined and secure user registration process.

1.1.1 Login

Registered users must log in to their Telegram account using their credentials. After logging in, they need to enter the bot and share their phone number to access the shopping system. There is a button labeled "Share Your Phone Number" in the T-Commerce Bot for this purpose.

1.1.2 Forget Password

As users have already been authenticated by Telegram , they need not provide a password to use this bot . For this , to handle "forget password " users can use Telegram settings .

To handle “Forgot Password” in telegram :

If you've forgotten your Telegram password, you can reset it by following these steps:

- Open the Telegram app and enter your phone number.
- When prompted for the password, select the "Forgot password?" option.
- Telegram will then send a verification code to your phone number via SMS.
- Enter the verification code in the app to verify your identity.
- You will then be prompted to create a new password for your account.

By following these steps, you should be able to reset your Telegram password and regain access to your account.

1.2 Dashboard

The **Dashboard** Module provides both customers and sellers with a centralized platform within the **T-Commerce** system to access their account information and monitor relevant data.

1. View Profile Details: When users(both sellers and customers) log in, clicking on “**Dashboard** “ button they can see their own profile information like their name, chat ID , created date and phone number.

[Note: If they need to update any of this information, they can easily do so right from their telegram account setting .]

2. Customer List (for Sellers): Sellers have access to a list of their customers.If admin clicks on “**users**” button , this shows who are registered and when ,who bought their products. It's a great way for sellers to keep track of their customer base .

1.3 Shopping System

The Shopping Management System Module is designed to facilitate seamless shopping experiences for buyers within the T-Commerce platform. To place an order they have to be registered .

1. **View Products:**

- Customers can explore a variety of products available on the platform.
- They can see product images, price, descriptions, and other relevant details.
- This feature is accessible through the Telegram bot. There is a button “ **Catalog** “ to handle this.

2. **Shopping Cart:**

- Customers can add products to their shopping cart.
- They can also remove items from the cart as needed.
- There is a button “ **Cart**“ to handle this.

3. **Modify Cart Contents:**

- Customers can view their shopping cart and make modifications.
- For example, they can update preferences, or remove items.
- This feature is accessible via the Telegram bot. There is a button “ **Clear Cart**“ to remove all items from the cart.
- .

4. **Place Orders:**

- Customers can place orders for the selected products.
- They can review their order details before confirming.
- Orders can be initiated through the Telegram bot or the website . After adding products to the cart , they will get “**Place Order**” to do this.

5. **Payment:**

After clicking on “**Place Order** “ button , a notification will be sent to the admin to manage the order . Admin can confirm or cancel orders. After confirmation , the bot automatically sends Link for payment . then clicking on the link , users can make payment . After successful payment , the admin and customers both will be notified.

1.4 Product Management

The Product Management Module in our T-Commerce platform gives sellers the tools they need to efficiently handle their product listings, categories, and incoming orders. Here's how it works:

1. Adding, Updating, and Deleting Products:

- Sellers can easily add new products to their catalog, including detailed descriptions, images, pricing, and other essential information.
- They can update existing product listings, making changes to information, prices, and inventory levels as needed.

2. Managing Categories:

- Sellers have the flexibility to organize their products into categories for easy navigation.
- They can create, edit, and delete categories to keep their catalog organized and user-friendly.

3. Viewing and Managing Orders:

- Sellers can view incoming orders and process them promptly.
- They have the ability to cancel orders if necessary, ensuring efficient order management.

For sellers using the Telegram bot, performing these actions is as easy as typing predefined commands like:

- **"add product"**
- **"modify product"**
- **"delete product"**
- **"add category"**
- **"modify category"**
- **"delete category"**
- **"Confirm order"**

- "Cancel order"

With these simple commands, sellers/admin can manage their product catalog and orders seamlessly, whether they're using the Telegram bot interface.

1.5 Notification Module

The Notification Module is integral to the T-Commerce platform, ensuring effective communication between sellers and customers through timely email notifications. By providing relevant updates and enabling prompt responses, the module contributes to an enhanced shopping experience and fosters positive relationships between sellers and customers. Here's how they work :

For Customers:

- Customers get notification through the Telegram Bot about their orders, like when they place an order, when it's confirmed, and if there are any updates (like processing or shipping).
- They also receive notification about making payment , payment confirmation , order status through the Telegram Bot .

For Sellers:

- Sellers get notification whenever something happens with their products or orders, like when someone places a new order.
- After making payment by the customers , sellers will also be notified about confirmation through the Telegram Bot.

Notifications are vital for keeping everyone informed and engaged in our T-Commerce platform. They make communication easy and help improve the overall shopping experience for customers and sellers alike.

2. Identifying Function Points

We use the client's perspective to identify functionality here. Hence, they are less granular.

Here are the two requested tables for the function points, based on the content extracted:

Table 1: Function Points by Functionality Type

FID	Functionality of the Software (As described by Client)	Type of Functionality
F1	Register and log in users via Telegram bot	External Input
F2	Edit account details and preferences through Telegram	External Input
F3	View categorized product details through Telegram bot	External Inquiry
F4	Add products to shopping cart and remove items	External Input
F5	Place orders and pay via Telegram bot	External Input
F6	Add, update, delete products and categories as a seller	Internal Files
F7	Access list of users who ordered products	Internal Files
F8	Provide product search functionality (including image-based search)	External Inquiry
F9	Personalized product recommendations	External Output

F10	Generate and manage sales reports	Internal Files
F11	Handle notifications for order confirmations, updates, and sales	External Output

Table 2: Unadjusted Function Points by Complexity

FID	Functionality of the Software (As described by Client)	Complexity
F1	Register and log in users via Telegram bot	Average
F2	Edit account details and preferences through Telegram	Simple
F3	View categorized product details through Telegram bot	Simple
F4	Add products to shopping cart and remove items	Average
F5	Place orders and pay via Telegram bot	Complex
F6	Add, update, delete products and categories as a seller	Average
F7	Access list of users who ordered products	Simple
F8	Provide product search functionality	Simple
F9	Personalized product recommendations	Complex
F10	Generate and manage sales reports	Complex
F11	Handle notifications for order confirmations, updates, and sales	Average

Table 3 : Unadjusted Function Points Table for Telegram Bot T-Commerce

FI D	Functionality of the Software	Complexity	Function Type	Weighting Factor
F1	Register and log in users via Telegram bot	Average	External Input	4
F2	Edit account details and preferences through Telegram	Simple	External Input	3
F3	View categorized product details through Telegram bot	Simple	External Inquiry	3
F4	Add products to shopping cart and remove items	Average	External Input	4
F5	Place orders and pay via Telegram bot	Complex	External Input	6
F6	Add, update, delete products and categories as a seller	Average	Internal Logical File	7
F7	Access list of users who ordered products	Simple	Internal Logical File	7
F8	Provide product search functionality	Simple	External Inquiry	3
F9	Personalized product recommendations	Complex	External Output	7
F10	Generate and manage sales reports	Complex	Internal Logical File	15

F11	Handle notifications for order confirmations, updates, and sales	Average	External Output	5
------------	--	---------	-----------------	---

Table 4 :Grouped Function Points by Type

Function Type	FID(s)	Complexity Levels	Weighting Factors	Total Count
External Input	F1, F2, F4, F5	Simple, Average, Complex	4, 3, 4, 6	17
Internal Logical File	F6, F7, F10	Simple, Complex	7, 7, 15	30
External Inquiry	F3, F8	Average, Complex	3, 3	6
External Output	F9, F11	Average, Complex	5, 7	12
Total UFP				65

Table 5 :Grouped Function Points by Type & Complexity

Function type	Simple	Average	Complex
Internal Logical File	7	10	15
External Interface File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

3. Degree of Influence (DI)

The following 14 factors influence the implementation aspects of the functionalities, aside from their inheritance of these complexities. These factors are the additional technical requirements that those functionalities must fulfill.

1. Data Communication
2. Distributed Functions
3. Performance
4. Heavily Utilized Hardware
5. High Transaction Rates
6. On-line Data Entry
7. Online Updating
8. End-user Efficiency
9. Complex Computations
10. Reusability
11. Ease of Installations
12. Ease of Operations
13. Portability
14. Maintainability/Facility Change

We will now estimate the influence of these factors by assigning a score between 1-5.

Table 6 :Complexity Influencing Factor & Degree of Influence

Complexity Influencing Factor	Degree of Influence
Data Communication	3
Distributed Functions	3
Performance	4
Heavily Utilized Hardware	1
High Transaction Rates	2
On-line Data Entry	3
Online Updating	3
End-user Efficiency	4

Complex Computations	4
Reusability	4
Ease of Installations	4
Ease of Operations	3
Portability	3
Maintainability/Facility Change	4
Total DI	45

4. Technical Complexity Factor(TCF)

Technical Complexity Factor (TCF) is a measure of the technical difficulty of a software project. It is used in software estimation to adjust the estimated size of the project based on the technical considerations involved. The TCF is typically calculated by assigning a score between 0 and 5 to each of a set of technical factors, such as the complexity of the system architecture, the use of new or unfamiliar technologies, and the level of integration required with other systems. The scores are then multiplied by weighted values for each factor, and the total of all calculated values is the TCF.

TCF is a key factor in the Use Case Points (UCP) estimation technique. UCP is a size estimation method that is based on the number and complexity of use cases in a system. The TCF is used to adjust the UCP estimate to account for the technical difficulty of the project.

$$TCF = 0.65 + 0.01 \sum_{i=1}^{14} F_i$$

Using this formula, TCF = 0.65 + 0.01 * 45 =1.1

5. Function Points

Function Points (FP) is a unit of measurement for software size. It is used to estimate the cost, effort, and schedule of software development projects. FP is calculated by counting the number of elementary processes in a software system. Elementary processes are the smallest units of functionality that are meaningful to the user.

FP is a widely used software size estimation method, and it is supported by a number of industry standards bodies, including the International Function Point User Group (IFPUG) and the Object Management Group (OMG).

$$FP = UFC \times TCF$$

So Function Points, $FP = UFP * TCP = 65 * 1.1 = 71.5$

The FP count can be used to estimate the cost, effort, and schedule of the software project by using historical data from previous projects. For example, if the average cost of developing a Function Point is \$100, then the estimated cost to develop a software project with an FP count of **71.5 would be \$7,150.**

6. Estimate Lines of Code (LoC) from Function Points

$$LOC = \text{Language Factor} * FP$$

Table 7 :Estimation LOC from Function Points

Language	Language Factor	Function Points	Lines of Code (LoC)
Assembly	320	71.5	$71.5 * 320 = 22,880$
C	128	71.5	$71.5 * 128 = 9,152$
Cobol	105	71.5	$71.5 * 105 = 7,508$
Fortran	105	71.5	$71.5 * 105 = 7,508$
Pascal	90	71.5	$71.5 * 90 = 6,435$
Ada	70	71.5	$71.5 * 70 = 5,005$

Java	55	71.5	$71.5 * 55 = 3,933$
JavaScript	47	71.5	$71.5 * 47 = 3,361$

We choose **JavaScript** as the language of choice due to ease of maintainability. So we estimate a total of **3.361 KLOC** for the software project.

Comparison with Real Project:

Table 8 :Comparison with real project

Category	File	Lines of Code (LoC)
Controllers	cart.js	115
	category.js	230
	index.js	65
	order.js	190
	payment.js	65
Views	index.ejs	85
	Error.ejs	30
	Callback.ejs	90
Models	Cart.js	46
	Category.js	8
	Order.js	35
	Product.js	15
	User.js	15
Bot	Query.js	115
	Message.js	85
	Bot.js	9

Menu	Keyboard.js	35
	Cart.js	115
	Category.js	230
	Order.js	190
	Payment.js	65
Products	product.js	260
	start.js	70
	User.js	45
Components	Button.jsx	16
	Button.css	45
	Card.js	36
Styles	cart.css	65
Cart Module	Cart.js	18
	Cart.css	6
Product Module	Product.js	12
	User.js	18
	Productlist.jsx	24
	ProductForm1.jsx	53
	Product.jsx	14
Database	Db.js	17

This version of real project sums to **3.011 KLOC**

Calculated KLOC using FP is **3.361 KLOC**

To calculate the **efficiency** of the project, we use the following formula:

$$\text{Efficiency} = \left(\frac{\text{Actual KLOC}}{\text{Calculated KLOC}} \right) \times 100$$

Where:

- **Actual KLOC** = 3.011 KLOC (the real total lines of code)
- **Calculated KLOC** = 3.361 KLOC (the estimated lines of code based on function points)

Now, let's calculate:

$$\begin{aligned} \text{Efficiency} &= \left(\frac{3.011}{3.361} \right) \times 100 \\ &= 89.59 \end{aligned}$$

The efficiency of the project, based on the comparison of actual KLOC to the calculated KLOC, is approximately **89.59%**.

7. Cost Estimation

The **COCOMO II model**, a widely used model for software cost estimation. It helps estimate the effort, time, and cost required to develop a software project based on factors like size, complexity, and type of software.

In the context of COCOMO II, the **mode** could refer to different types of software projects, and each mode has specific parameters (like the constants and exponents) that are applied based on the project type.

Here's a breakdown of the modes:

1. Organic:

- Used for relatively simple, small projects.
- Typically, small teams with good experience and familiarity with the software domain.
- Examples: Small business applications, independent software development.

2. Semi-Detached:

- Used for moderately complex projects.
- The project involves a mix of experienced and less experienced teams.
- Examples: Medium-sized software development in organizations.

3. Embedded:

- Used for highly complex projects.
- These projects have stringent constraints (e.g., real-time systems, safety-critical systems).
- Examples: Aerospace, defense, or other systems requiring specialized hardware integration.

Table 9 :COCOMO II codes

Mode	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

- **a, b, c, d:** These constants are used in the COCOMO II formula to estimate the effort and cost required. The **a** value represents the scaling factor based on the project size, **b** reflects the learning curve, and so on.
- The different models (Organic, Semi-Detached, and Embedded) apply these constants based on the complexity of the project.

We selected Organic Mode for the software project. Because the SPL2 process resembles it more and mainly because the need for strict industry requirements is nonexistent here - it is an organic environment.

So, Effort = $a \cdot (KLOC)^b = 2.4 \cdot (3.361)^{1.05} = 8.57$

$$\text{Duration} = c * (\text{Effort})^d = 2.5 * (8.57)^{0.38} = 5.656$$

$$\text{Person Required} = \text{Effort} / \text{Duration} = 8.57 / 5.656 = 1.515 \sim 2$$

Software Effort Distribution for RUP/MBASE (Person-Months)

Table 10 : Software effort distribution

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.2	0.5	0.1
Environment/CM	0.0	0.1	0.2	0.0
Requirements	0.1	0.3	0.4	0.0
Design	0.1	0.5	0.8	0.0
Implementation	0.0	0.2	1.6	0.1
Assessment	0.0	0.2	1.1	0.2
Deployment	0.0	0.0	0.1	0.2

Table 11: Final Normalized Table (On a Scale of 1)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.25	0.08	0.11	0.17
Environment/CM	0.0	0.04	0.04	0.0
Requirements	0.25	0.12	0.09	0.0
Design	0.25	0.20	0.17	0.0
Implementation	0.0	0.08	0.34	0.17
Assessment	0.25	0.08	0.23	0.33
Deployment	0.0	0.0	0.02	0.33

References

[1] <https://people.cs.ksu.edu/~padmaja/Project/CostEstimate>

[2] SPL-2 Report

<https://docs.google.com/document/d/1GpIM6-onkZNBcLNQLGtypXl9eOhPcJv9ps77dBVvTsk/edit?tab=t.0>

[3] SPL-2 SRS :

https://docs.google.com/document/d/1QMrasXiBUeMG90OpnYh-WfVtdEFWB0zlE0aBzMde_l4/edit?tab=t.0

[4] SPL-2 Project Link: https://github.com/AmitRoy01/T_commerce