
Requirement & Function Size of SPL 2

for

IIT Library 2.0

Submitted To

Dipok Chandra Das
Assistant Professor
Institute of Information Technology (IIT)
Noakhali Science and Technology University

Submitted by

Prosanto Deb (ASH1925005M)
Abdullah Alif (ASH1925009M)
Ratno Kumer Tripura (ASH1825042M)

Course Title: Software Metrics Lab

Course Code: SE 3204

Date of Submission: 05/03/2023

Table of Contents

Table of Contents	ii
1. Project Information	1
2. Requirement Size	1
2.1 Stakeholders	2
2.2 Functional & Non-functional	2
2.3 Activity Diagram	3
2.4 Total Information	4
3. Function Size	4
3.1 Function Points.....	5
3.2 Technical Complexity Factor	6

Table of Figures

No table of figures entries found.

1. Project Information

Project Name	IIT LIBRARY 2.0
Supervised By	Dipok Chandra Das Assistant Professor Intitute of Information Technology(IIT) Noakhali Science and Technology University
Team Members	Prosanto Deb (ASH1925005M)
	Abdullah Alif (ASH192509M)
	Ratno Kumer Tripura (ASH1825042M)
GitHub Link	IIT-Library-2.0
SRS Document	Software Requirement & Specification File

2. Requirement Size

Definition: Requirements and specification documents generally combine text, graphs, and special mathematical diagrams and symbols. The nature of the presentation depends on the particular style, method, or notation used. When measuring code or design size, you can identify atomic entities to count (lines, statements, bytes, classes, and methods, for example). However, a requirements or specification document can consist of a mixture of text and diagrams. For example, a use case analysis may consist of a UML use case diagram along with a set of use case scenarios that may be expressed as either text or as UML activity diagrams. Because a requirements analysis often consists of a mix of document types, it is difficult to generate a single size measure

Process

- There are obvious atomic elements like number of pages that can be counted.
- **Use case diagrams:** Number of use cases, actors, and relationships of various types.

- **Use case:** Number of scenarios, size of scenarios in terms of steps, or activity diagram model elements.
- **Domain model** (expressed as a UML class diagram): Number of classes, abstract classes, interfaces, roles, operations, and attributes.

Values

2.1 Stakeholders

Metric Type	Metric	Measurement	
Stakeholders	Number of stakeholders	Type	Quantity
		Acquirer	5
		Supplier	1
		Change Control Board	1

2.2 Functional & Non-functional

Metric Type	Metric	Measurement
Functional & non-functional requirements	Number of functional requirements	7
	Number of non-functional Requirements	5
Requirements gathering	Number of persons interacted	3
	Number of interviews taken	4
Use case diagram	Number of use cases	29
	Number of actors	7
	Number of relations between use cases	21
	Number of relations between use case and actors	30
	Number of relations per use cases	1.77
Use case description	Number of use case descriptions	28
	Number of actors per use case (Average)	2.55

2.3 Activity Diagram

Metric Type	Properties	Metric	
Activity diagram	Title	Number of steps	Number of branching actions
	Send Registration Request	7	2
	Borrow Request	10	2
	Extend Book Period	10	2
	Access Electronic Copy	8	2
	Request New Book	9	2
	Check Fine	6	2
	Search Book (By User)	6	2
	Login	7	2
	Admin Login	6	2
	Send Emails	5	2
	Logout	5	2
	Validate Borrower ID	5	2
	Authenticate User and Librarian	4	2
	Calculate Fine	5	4
	Show Book	4	2
	Borrower Receipt Generate	4	2
	Report Generate	4	2
	Watch New Book Requests	6	2
	Send Request for Approval	6	2
	Search Book (By Librarian)	6	2
	Add New Book	6	2
	Update Book Information	6	2
	Delete Book	5	2
	Remove Book	5	2
	Approve Request	5	2
	Add Librarian	7	2
	Delete Librarian	7	2

	Delete User	7	2
	Update Password	9	2

2.4 Total Information

Metric Type	Metric	Measurement
Activity diagram	Total number of activity diagrams	28
	Total number of Steps	180
	Total number of branching actions	60
	Average Steps per activity diagram	6.2
	Average branching actions per activity diagram	2.06

3. Function Size

Albrecht's effort estimation method was largely based on the notion of FPs. As their name suggests, FPs are intended to measure the amount of functionality in a system as described by a specification. We can compute FPs without forcing the specification to conform to the prescripts of a particular specification model or technique.

To compute the number of FPs we first compute an unadjusted function point count (UFC).

- **External inputs:** Those items provided by the user that describe distinct application-oriented data (such as file names and menu selections). These items do not include inquiries, which are counted separately.
- **External outputs:** Those items provided to the user that generate distinct application-oriented data (such as reports and messages, rather than the individual components of these).
- **External inquiries:** Interactive inputs requiring a response.
- **External files:** Machine-readable interfaces to other systems.
- **Internal files:** Logical master files in the system.

Next, each item is assigned a subjective "complexity" rating on a three point ordinal scale: simple, average, or complex. Then, a weight is assigned to the item, In theory, there are 15 different varieties of items (three levels of complexity for each of the five types), so we can compute the UFC by multiplying the number of items in a variety by the weight of the variety and summing over all 15.

3.1 Function Points

Function Points	Quantity	Name	Complexity	Complexity Weight
External Inputs	3	Book information	Complex	6
		User information	Complex	6
		Librarian information	Average	4
External Outputs	9	Book availability	Average	5
		Number of bookings	Simple	4
		Number of book requests	Simple	4
		Number of borrowed books	Average	5
		Fine Amount	Complex	7
		Due Date	Average	5
		Top 5 Books (In Quantity)	Complex	7
		Top 5 Borrowed Books	Complex	7
		Top 5 Borrowers	Complex	7
External Inquiries	7	Invalid input message	Average	4
		Registration approved message	Complex	6
		Mail sent message	Complex	6
		Update information message	Average	4
		Book added message	Simple	3
		Book not found message	Complex	6
		Borrow duration extend message	Complex	6
External Files	2	Cover pictures	Complex	15
		PDF books	Complex	15
Internal Files	1	Library Database	Average	7
Unadjusted Function Point (UFC)				139

To complete our computation of FPs, we calculate an adjusted function point count, FP, by multiplying UFC by a technical complexity factor, TCF. Each component or sub factor is rated from 0 to 5, where 0 means the sub factor is irrelevant, 3 means it is average, and 5 means it is essential to the system being built.

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

$$FP = UFC \times TCF$$

3.2 Technical Complexity Factor

No.	Complexity Factors	Rating (0-5)
F1	Reliable backup and recovery	5
F2	Data communications	2
F3	Distributed functions	0
F4	Performance	5
F5	Heavily used configuration	3
F6	Online data entry	3
F7	Operational ease	5
F8	Online update	4
F9	Complex interface	4
F10	Complex processing	5
F11	Reusability	5
F12	Installation ease	5
F13	Multiple sites	2
F14	Facilitate change	4
Total		52

Technical Complexity Factor (TCF) = $0.65 + 0.01 * 52 = 1.17$

Function Point (FP) = Unadjusted Function Point (UFC) * Technical Complexity Factor (TCF)

$$= 139 * 1.17 = \mathbf{162.63}$$

Here if we assign 1day to implement 1 function points that the project will take more than 5 months.