SAPIENZA UNIVERSITÀ DI ROMA

February 3, 2024

Laboratory of Advanced Programming: Function Point Analysis

by

Emilio Martino Emanuele Roncioni Eric Welmillage
Giuseppe Fosci Alessio Rago

Abstract:

This document presents a detailed analysis of function points as an integral part of the Laboratory of Advanced Programming course project and a fundamental part of the Software Engineering course.

In this study, we aim to measure how well a software system works using function points, a common method for sizing and estimating software. Our system has two Internal Logical Files (ILFs) called 'User' and 'Recommendations,' which store data about user profiles and recommendations. We also have another ILF called 'SpotifySession,' which holds data about user sessions within the Spotify app.

The main advantages of the FP technique, are being sufficiently objective and sufficiently independent of the technology used in development; in fact, in the "final step" we will use a table that based on the programming language will give us an objective indication of the number of lines of code.

1 Tables used to calculate FP

In this section we briefly recall the basic elements of FP calculation, which will be based on the tables below (the values in these tables are taken from the examples of the exercises in the Software Engineering course).

Table 1: ILF/EIF complexity

Ret/Det	**1-19 Det**	**20-50 Det**	**51+ Det**
1 Ret	Low (7/5)	Low (7/5)	Medium (10/7)
2-5 Ret	Low (7/5)	Medium (10/7)	High (15/10)
6+ Ret	Medium (10/7)	High (15/10)	High (15/10)

Table 2: Transaction Compexity: EI

FTR/DET	**1-4 DET**	**5-15 DET**	**16+ DET**
0 – 1 FTR	Low (3)	Low (3)	Medium (4)
2 FTR	Low (3)	Medium (4)	High (6)
3+ FTR	Medium (4)	High (6)	High (6)

Table 3: Transaction complexity: EO/EQ

FTR/DET	**1-5 DET**	**6-19 DET**	**20+ DET**
0 – 1 FTR	Low (4/3)	Low (4/3)	Medium (5/4)
2-3 FTR	Low (4/3)	Medium (5/4)	High (7/6)
4+ FTR	Medium (5/4)	High (7/6)	High (7/6)

2 Project Description

Initially, the project involves creating separate microservices with connections between them that will be implemented later. Ideally, the system should communicate externally only through the external interfaces (1, 2) in response to inputs and through the API call service (5) to initiate output. All other communications occur within an internal virtual network connecting the various containers.

3 Requirements - User Stories

Microservices

1. External Interface, API
2. External Interface, Browser
3. Business Logic
4. Machine Learning Engine /
5. API Call
6. Database

Summarizing communications:

- Interfaces 1 and 2 must independently communicate with Business Logic (3), likely through REST API.
- Elements 4 and 5 must also communicate with Business Logic. Since this represents a potential single point of failure, redundancy at the Business Logic node level must be ensured.
- The API call service (5) must be the only element in the network writing to the databases.

External Interface, API: 6 Python Flask

1. Can request recommendations via API key. (EO)
2. Can request all recommendations received in the last week in a structured list. (EQ)
3. Can check the number of API calls made daily and how many are remaining on their plan. (EO)
4. Can obtain information about their account. (EQ)
5. A user automatically refresh its tokens every day (EO)
6. Can view a landing page introducing the service. (EQ)
7. Can sign up for the service. (EI)
8. Can view a homepage containing user information present in the system. (EQ)
9. Can obtain their API key to make API calls. (EO)
10. Can receive a recommendation. (EO)
11. Can listen to a snippet of their recommendation via Spotify. (EQ)
12. Can listen to their recommendation on Spotify via a link. (EQ)
13. Can view remaining daily recommendations. (EO)
14. Can rate a received recommendation to get more daily recommendations. (EI)
15. Can suggest a song to be associated with another to get more daily recommendations. (EQ)
16. A user machine can receive information about remaining daily service usage. (EI)
17. A user machine can receive information about a recommendation made by

delegating to the machine learning engine (perhaps also with user data). (EI-ML)

18. A user machine can receive information about previous recommendations made by making a call to the DBMS. (EI)

19. A user machine can receive Spotify data on received recommendations by delegating a call to the API call service. (EI)

20. A user machine can add a user to the database by performing a check on the DBMS. (EI)

21. A user machine can remove a user from the database by performing a check on the DBMS. (EI)

22. A user machine can modify the number of calls allowed by a user. (EI)

23. A user machine can add information to the dataset by delegating to the business logic to inform the machine learning engine of new available data (divisible into 2 for recommendations and ratings). (EI)

24. A user machine can authenticate a user via a key. (EI)

25. A user machine can receive data about a user by delegating to the business logic service a call to the DB. (EI)

In this study, we aim to measure how well a software system works using function points, a common method for sizing and estimating software. Our system has two Internal Logical Files (ILFs) called ’User’ and ’Recommendations,’ which store data about user profiles and recommendations. We also have another ILF called ’SpotifySession,’ which holds data about user sessions within the Spotify app.

ILF and EIF Complexity:
2 ILFs (Low complexity) → 14 FP
1 EIF (Low complexity) → 7 FP

EI (External Input) Complexity

User Story	FTR (File Type Referenced)	DET (Data Element Types)	Complexity	FP
US7 - (User)SignUp	1	3	Low	3
US14 (User Racc.)	3	6	High	6
US16 (Remaining Tokens)	1	2	Low	3
U17 - (ML-Racc.)	1	3	Low	3
US18 - (Previous Racc.)	2	3	Low	3
US19 (Racc. API call)	1	3	Low	3
US20 (Add User)	1	4	Low	3
US21 (Remove User)	1	1	Low	3
US23 (Add data ML)	1	20	Medium	4
US24 (User Auth)	1	1	Low	3
US25 (Receive data)	1	4	Low	3

Total: 11 EIs (Low complexity) → 37 FP

EO (External Output) Complexity

User Story	FTR (File Type Referenced)	DET (Data Element Types)	Complexity	FP
US1 (Recommendations)	1	3	Low	4
US3 (Check token)	1	1	Low	4
US5 (Refresh Tokens)	1	1	Low	4
US9 (Api Key)	1	1	Low	4
US10 (Recommendations)	1	3	Low	4
US13 (Daily Recommendations)	1	3	Low	4

Total: 6 EOs (Low complexity) → 24 FP

EQ (External Inquiry) Complexity

User Story	FTR (File Type Referenced)	DET (Data Element Types)	Complexity	FP
US2 (Recommendations)	1	20	High	6
US4 (Obtain information)	1	4	Low	4
US6 (Landing page)	1	4	Low	0
US8 (User info)	1	4	Low	4
US11 (Racc. on Spotify)	1	4	Low	4
US12 (Racc. on Spotify via link)	1	4	Low	4

Total: 6 EQs (Low and High complexity) $\rightarrow 22$ FP