

Session 2

Effort and cost estimate for the Railway Reservation System using Function Point Analysis and COCOMO

This session turns the scope statement of Session 1 into numbers: how big the Railway Reservation System (RRS) is, how many person-months it needs, how long it will take, how many people to hire, and what it will cost. The manual asks for “any estimation technique”. We use two that fit together: Function Point Analysis (FPA) to measure size from the scope, and the basic COCOMO model to convert size into effort, duration, and cost. A third, quicker method (expert judgement) is used as a cross-check. Session 13 cites these figures when comparing the estimate with the real program, so keep every number exactly as computed here.

Objectives

✕ Do not copy. Read for understanding and the viva

- Count unadjusted function points for the eight RRS modules from the scope statement.
- Rate the 14 general system characteristics and compute the value adjustment factor.
- Convert adjusted function points to lines of code and apply basic COCOMO (organic mode).
- Derive duration, average staff, and cost in rupees from the effort figure.
- Cross-check the model estimate against an expert judgement estimate and explain the difference.

Problem Statement

■ Write in lab record

Session 2: Estimate the effort and cost required to build the above software. Use any estimation technique.

Concept

✕ Do not copy. Read for understanding and the viva

Why estimate before design

The project plan (manual section 1.2) needs a delivery deadline, a resource requirement, and a budget before the SRS exists. All three come from an estimate of size and effort. An estimate made from scope is rough (plus or minus 25 percent is normal), so it is revised after the SRS and again after design. The value of this session is the method, which stays the same at each revision.

Function Point Analysis

FPA measures size from what the software does for its users, not from code. Each function in the scope is classified as one of five types and rated simple, average, or complex.

Type	Meaning	Simple	Average	Complex
External Input (EI)	Data enters the system and updates a file (a form, a transaction)	3	4	6
External Output (EO)	Derived data leaves the system (a report, a receipt, a message with computed values)	4	5	7
External Inquiry (EQ)	A request that retrieves data without deriving or updating it (a lookup)	3	4	6
Internal Logical File (ILF)	A logical group of data maintained inside the system (a master table)	7	10	15
External Interface File (EIF)	A logical group of data used by the system but maintained outside it	5	7	10

Complexity depends on the number of data elements and files touched. For a scope-level count use a simple rule: one form with a few fields is simple; a form that touches two files is average; a form that touches three or more files or has many rules (route definition, booking with up to 6 passengers and seat allocation) is complex. Each ILF is counted once, in the module that maintains it.

Unadjusted function points (UFP) is the sum of weights. Then 14 general system characteristics (GSC) are each rated 0 to 5 and the value adjustment factor is VAF equals 0.65 plus 0.01 times the sum of ratings. Adjusted function points AFP equals UFP times VAF.

From function points to lines of code

COCOMO needs size in thousands of lines of code (KLOC). Published backfiring tables give an average lines-per-function-point for each language. For Python we use 60 LOC per FP, a conservative figure for a beginner team writing readable code with docstrings.

Basic COCOMO

Basic COCOMO (Boehm, 1981) has three modes. Organic mode is for small teams, familiar problems, and flexible requirements, which describes an MCA project team building a reservation system. The formulas are:

- Effort E equals 2.4 times KLOC to the power 1.05, in person-months (PM)
- Duration D equals 2.5 times E to the power 0.38, in calendar months
- Average staff equals E divided by D

Semi-detached mode (3.0, 1.12, 2.5, 0.35) and embedded mode (3.6, 1.20, 2.5, 0.32) give higher effort and are used for larger or tightly constrained systems. The exponent above 1 means effort grows faster than size, because communication overhead grows with the team.

Expert judgement

An experienced developer looks at each module and states a figure from memory of similar work. It is fast and captures things a model cannot see, but it is only as good as the expert. Using it alongside a model and explaining any gap is normal practice.

Estimation

■ Write in lab record

1. Function count by module

Functions are taken from section 4 of the Session 1 scope statement.

Module	Function	Type	Complexity	FP
1 User Management	Register user	EI	Average	4
1 User Management	Login	EI	Simple	3
1 User Management	Update profile	EI	Simple	3
1 User Management	Change password	EI	Simple	3
1 User Management	View profile	EQ	Simple	3
1 User Management	User file	ILF	Simple	7
2 Train and Schedule	Add train	EI	Average	4
2 Train and Schedule	Add station	EI	Simple	3
2 Train and Schedule	Define route (stop sequence, times, distance)	EI	Complex	6
2 Train and Schedule	Add coach with seats	EI	Simple	3
2 Train and Schedule	Set fare per km per class	EI	Simple	3
2 Train and Schedule	Create schedule for run date	EI	Average	4
2 Train and Schedule	Update schedule status (cancelled, delayed)	EI	Average	4
2 Train and Schedule	Station file	ILF	Simple	7
2 Train and Schedule	Train file (Train, Route, Coach, Seat)	ILF	Complex	15
2 Train and Schedule	Schedule file	ILF	Simple	7
2 Train and Schedule	Fare file	ILF	Simple	7
3 Search and Availability	Search trains by source, destination, date	EQ	Complex	6

Module	Function	Type	Complexity	FP
3 Search and Availability	Check availability per class	EQ	Average	4
3 Search and Availability	Fare enquiry	EQ	Simple	3
3 Search and Availability	List stations	EQ	Simple	3
4 Booking	Create booking (passengers, seat allocation, PNR, waitlist)	EI	Complex	6
4 Booking	Show and print ticket	EO	Average	5
4 Booking	PNR status enquiry	EQ	Average	4
4 Booking	Booking file (Booking, Passenger)	ILF	Average	10
5 Cancellation and Refund	Cancel booking (release seats, promote waitlist)	EI	Complex	6
5 Cancellation and Refund	Refund statement (amount by refund rule)	EO	Average	5
5 Cancellation and Refund	Refund status enquiry	EQ	Simple	3
5 Cancellation and Refund	Refund file	ILF	Simple	7
6 Payment	Initiate payment	EI	Average	4
6 Payment	Payment gateway callback (confirm or fail)	EI	Average	4
6 Payment	Payment receipt	EO	Simple	4
6 Payment	Payment file	ILF	Simple	7

Module	Function	Type	Complexity	FP
6 Payment	Payment Gateway transaction data	EIF	Simple	5
7 Notification	Booking confirmation message	EO	Simple	4
7 Notification	Cancellation message	EO	Simple	4
7 Notification	Schedule change alert	EO	Average	5
7 Notification	Notification log	ILF	Simple	7
7 Notification	Notification Service delivery status	EIF	Simple	5
8 Reports	Occupancy report per train per schedule	EO	Complex	7
8 Reports	Revenue report by date range	EO	Average	5
8 Reports	Cancellation and refund report	EO	Average	5

2. Unadjusted function points

Each cell shows the number of functions and, in brackets, their function points.

Module	EI	EO	EQ	ILF	EIF	UFP
1 User Management	4 (13)	0	1 (3)	1 (7)	0	23
2 Train and Schedule Management	7 (27)	0	0	4 (36)	0	63
3 Search and Availability	0	0	4 (16)	0	0	16
4 Booking	1 (6)	1 (5)	1 (4)	1 (10)	0	25
5 Cancellation and Refund	1 (6)	1 (5)	1 (3)	1 (7)	0	21
6 Payment	2 (8)	1 (4)	0	1 (7)	1 (5)	24
7 Notification	0	3 (13)	0	1 (7)	1 (5)	25
8 Reports	0	3 (17)	0	0	0	17
Total	15 (60)	9 (44)	7 (26)	9 (74)	2 (10)	214

Check by type: $60 + 44 + 26 + 74 + 10 = 214$. Check by module: $23 + 63 + 16 + 25 + 21 + 24 + 25 + 17 = 214$.

UFP = 214

3. General system characteristics

No.	Characteristic	Rating (0 to 5)	Reason
1	Data communications	4	Web application; every function goes over HTTPS
2	Distributed data processing	2	One server, but payment and notification are remote calls
3	Performance	4	Search within 3 seconds under 500 concurrent users
4	Heavily used configuration	2	Ordinary web server and database server
5	Transaction rate	4	Peak booking load when new schedules open
6	Online data entry	5	All input is interactive; there is no batch entry
7	End-user efficiency	3	Passengers are untrained public users; screens must be simple
8	Online update	4	Seat availability updates in real time on every booking and cancellation
9	Complex processing	2	Refund rules and waitlist promotion; no heavy computation
10	Reusability	2	Payment and notification components can be reused by other railway systems
11	Installation ease	1	Installed once on one server
12	Operational ease	3	Daily schedule creation and backups need an operator
13	Multiple sites	2	Counters at many stations use the same server

No.	Characteristic	Rating (0 to 5)	Reason
14	Facilitate change	2	Fares and refund rules change through data, not code
	Total degree of influence	40	

$$\text{VAF} = 0.65 + 0.01 \times 40 = 0.65 + 0.40 = 1.05$$

4. Adjusted function points and size

$$\text{AFP} = \text{UFP} \times \text{VAF} = 214 \times 1.05 = 224.7, \text{ rounded to } 225 \text{ FP}$$

$$\text{Size} = 225 \text{ FP} \times 60 \text{ LOC per FP} = 13,500 \text{ LOC} = 13.5 \text{ KLOC}$$

5. Effort, duration, and staff (basic COCOMO, organic mode)

Effort:

```

1 E = 2.4 × (KLOC)^1.05
2   = 2.4 × (13.5)^1.05
3
4   log10(13.5)           = 1.1303
5   1.1303 × 1.05         = 1.1868
6   antilog10(1.1868)    = 15.38
7
8 E = 2.4 × 15.38 = 36.9 person-months

```

Duration:

```

1 D = 2.5 × (E)^0.38
2   = 2.5 × (36.9)^0.38
3
4   log10(36.9)           = 1.5671
5   1.5671 × 0.38         = 0.5955
6   antilog10(0.5955)    = 3.94
7
8 D = 2.5 × 3.94 = 9.85 calendar months, rounded to 10 months

```

Average staff:

```

1 Staff = E / D = 36.9 / 9.85 = 3.75, rounded up to 4 persons

```

Productivity check: 13,500 LOC / 36.9 PM = 366 LOC per person-month, which is within the 300 to 500 range expected for an organic project in a high-level language.

6. Cost

Rate assumed: Rs 60,000 per person-month, the fully loaded cost (salary, workspace, equipment) of a junior developer in an Indian software services company.

Item	Calculation	Amount (Rs)
Development effort	36.9 PM × Rs 60,000	22,14,000
Server hosting for 12 months	Rs 6,000 × 12	72,000
Payment gateway and SMS gateway setup	Fixed	38,000
Development tools and licences	Fixed	40,000
Total estimated cost		23,64,000

Estimated cost of the RRS: Rs 23,64,000 (about Rs 23.6 lakh), delivered in 10 months by a team of 4.

7. Team plan

Role	Persons	Months engaged
Project lead and analyst (also designs and reviews)	1	10
Developers	2	10
Tester and documentation	1	7 (from month 4)

Person-months engaged: $10 + 20 + 7 = 37$, which matches $E = 36.9$ PM.

8. Cross-check by expert judgement

Three developers with reservation-system experience were asked to estimate each module in person-months. Their averages:

Module	Expert estimate (PM)	FPA share (PM)
1 User Management	3	4.0
2 Train and Schedule Management	6	10.9
3 Search and Availability	3	2.8
4 Booking	6	4.3
5 Cancellation and Refund	3	3.6
6 Payment	4	4.1
7 Notification	2	4.3
8 Reports	3	2.9
Integration testing and documentation	4	included above
Total	34	36.9

The FPA share column distributes $E = 36.9$ PM in proportion to each module's UFP (for example module 2: $63 / 214 \times 36.9 = 10.9$).

Comparison: expert judgement gives 34 PM and Rs 20,40,000 for development effort against 36.9 PM and Rs 22,14,000 from COCOMO, a gap of 8 percent. The experts rate Booking higher than FPA does because seat allocation and waitlist logic are hard to get right even though they are one function; they rate Train and Schedule Management lower because it is mostly simple data entry forms. Both figures fall within the plus or minus 25 percent expected at the scope stage. We adopt the COCOMO figure of 36.9 PM because it is the larger and safer one, and record the expert figure as the lower bound.

9. Summary of the estimate

Quantity	Value
Unadjusted function points	214
Value adjustment factor	1.05
Adjusted function points	225
Size	13,500 LOC (13.5 KLOC)
Effort	36.9 person-months
Duration	9.85 months (10 months)
Average staff	4
Development cost	Rs 22,14,000
Total cost including resources	Rs 23,64,000
Expert judgement cross-check	34 PM, Rs 20,40,000

Viva Questions

✕ Do not copy. Read for understanding and the viva

- **Q:** Why count function points instead of lines of code directly? **A:** Function points can be counted from the scope before any code exists, and they do not depend on the language.
- **Q:** What is the difference between an EO and an EQ? **A:** An EO contains derived or computed data (a report, a refund amount); an EQ only retrieves stored data (PNR status).
- **Q:** Why is the Train file counted as complex? **A:** It groups Train, Route, Coach, and Seat, many data elements across several record types.
- **Q:** What does VAF do? **A:** It scales the raw count by up to plus or minus 35 percent for technical factors like performance, online update, and multiple sites.
- **Q:** Why organic mode? **A:** Small team, familiar application type, and flexible requirements; embedded mode is for real-time or safety-critical systems.
- **Q:** Why is the exponent in the effort formula more than 1? **A:** Effort grows faster than size because communication and integration overhead rise with team size.
- **Q:** Why is duration not simply effort divided by team size? **A:** Adding people does not shorten work proportionally; COCOMO derives duration from effort with an exponent of 0.38.
- **Q:** What would change if the team used C instead of Python? **A:** LOC per FP would roughly double (about 128), so KLOC, effort, and cost would rise.
- **Q:** How accurate is this estimate? **A:** Plus or minus 25 percent at the scope stage; it is revised after the SRS and after design.

Common Mistakes

× Do not copy. Read for understanding and the viva

- Picking a KLOC figure from the air (such as 75 KLOC) with no function count behind it; the examiner asks where the number came from.
- Counting the same ILF in several modules; each file is counted once, in the module that maintains it.
- Using intermediate COCOMO cost drivers with basic COCOMO constants, or mixing constants from two modes.
- Rounding too early; keep two decimals until the final line, then round.
- Quoting cost in dollars with no rate; state the rate per person-month in rupees and show the multiplication.
- Leaving out duration and staff; effort alone does not give a delivery date.

Session Summary

■ Write in lab record

Submit in the lab record:

- The problem statement copied from the manual.
- The function count table (section 1) and the UFP summary table (section 2) totalling 214.
- The 14 GSC ratings with reasons, VAF 1.05, AFP 225, and size 13,500 LOC.
- The COCOMO working for E, D, and staff with logarithm steps shown.
- The cost table with the stated rate, and the expert judgement comparison.
- The summary table (section 9); Session 13 will compare actual code size against it.

 Edit this page

< Previous
Session 1

Next >
Session 3