# **COCOMO COST DRIVERS AND MULTIPLIERS FOR FINVERSE PROJECT**

# 1. Introduction

The Constructive Cost Model (COCOMO) uses cost drivers and multipliers to adjust effort estimates based on product, hardware, personnel, and project attributes.

# 2. Application to FinVerse Project

Based on the FinVerse project plan (scope, team, schedule, and risks), the following ratings are chosen for each of the 15 cost drivers. Each entry shows the chosen rating, the corresponding multiplier, and a brief justification.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cost Driver** | **Chosen Rating** | **Multiplier** | **Justification** |
| RELY (Required Reliability) | Nominal | 1.00 | Demo trading platform (mock data), no real-money transactions — standard requirement. |
| DATA (Database Size) | Low | 0.94 | Small database limited to demo users, portfolios, and mock transactions. |
| CPLX (Product Complexity) | High | 1.15 | Includes mock API integration for forex/stocks, wallet simulation, portfolio tracking, admin panel. |
| TIME (Execution Time Constraint) | Nominal | 1.00 | No strict hard real-time constraints; acceptable latencies for demo data. |
| STOR (Main Storage Constraint) | Nominal | 1.00 | Storage needs are modest with mock data; no large datasets. |
| VIRT (Virtual Machine Volatility) | Nominal | 1.00 | Standard deployment environment with typical VM stability. |
| TURN (Computer Turnaround Time) | Nominal | 1.00 | No extreme turnaround-time requirements for user interactions. |
| ACAP (Analyst Capability) | Nominal | 1.00 | Project team includes students with adequate analysis skills; no senior analysts. |
| AEXP (Applications Experience) | Low | 1.13 | Limited prior experience building trading/finance apps; learning required. |
| PCAP (Programmer Capability) | Nominal | 1.00 | Team programmers have standard academic/project experience. |
| VEXP (Virtual Machine Experience) | Nominal | 1.00 | Team has typical experience deploying to VMs/containers. |
| LEXP (Programming Language Experience) | Nominal | 1.00 | Team is familiar with chosen languages/frameworks at a normal level. |
| MODP (Modern Programming Practices) | High | 0.91 | Team uses modern frameworks and practices (version control, code reviews, modular design). |
| TOOL (Use of Software Tools) | High | 0.91 | Use of modern IDEs, build tools, and project management tools is expected. |
| SCED (Required Development Schedule) | High | 1.04 | 3-month deadline is fairly tight for the scope — slight schedule compression. |

### ****FinVerse Project Analysis****

**Size & Complexity:**

* The app has APIs, wallets, admin panel, and trading simulation — moderately complex.
* But no real financial transactions, so constraints are lower than real trading apps.

**Team Experience:**

* Student team with moderate programming and analysis skills (not industry veterans).
* Experience with finance/trading apps is limited (AEXP = Low).

**Constraints:**

* No strict real-time execution constraints (TIME = Nominal).
* Database and storage needs are small.
* Reliability requirements are not very high (demo project, no money at risk).

**Environment:**

* Web application only, no hardware/software tight coupling.

### CONCLUSION:

The **FinVerse Web App project best fits the** Semidetached **mode**:

* It’s more complex than an Organic project (due to API integration, admin panel, trading logic).
* The team has mixed experience (somewhat new to trading systems, but with programming knowledge).
* Constraints are not strict enough to classify as Embedded.

1. **Effort estimation (semi detached)**

**∵Effort = a x size b x EAF(effort estimation factor)**

Size = 20 KLOC

EAF = ∏ (Multiplier of Cost Driver)

=1.00x0.94x1.15x1.00x1.00x1.00x1.00x1.00x1.13x1.00x1.00x1.00x0.91x0.91x1.04

=1.052

Effort = 3.0 x 20 1.12 x1.052

= 90.426 person-months

**∵Duration = c x Effort d**

Duration = 2.5 x 90.426 0.35

= 12.092 months

**∵Staff = Effort/Duration**

Staff = 90.426/12.092

= 7.5 people

≈ 8 people