**COCOMO**

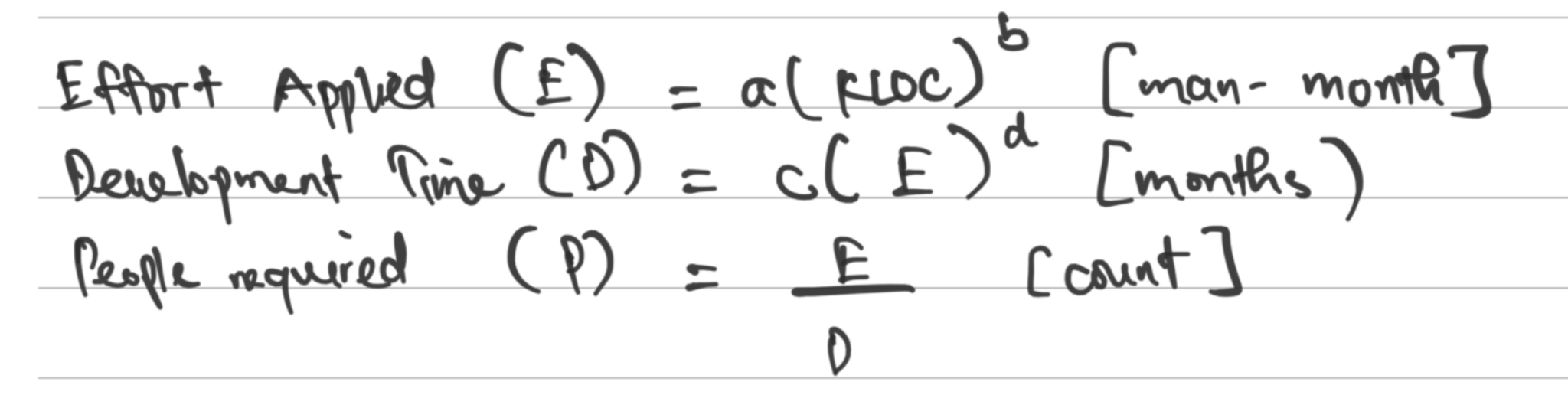
# 

# 1. Mở đầu

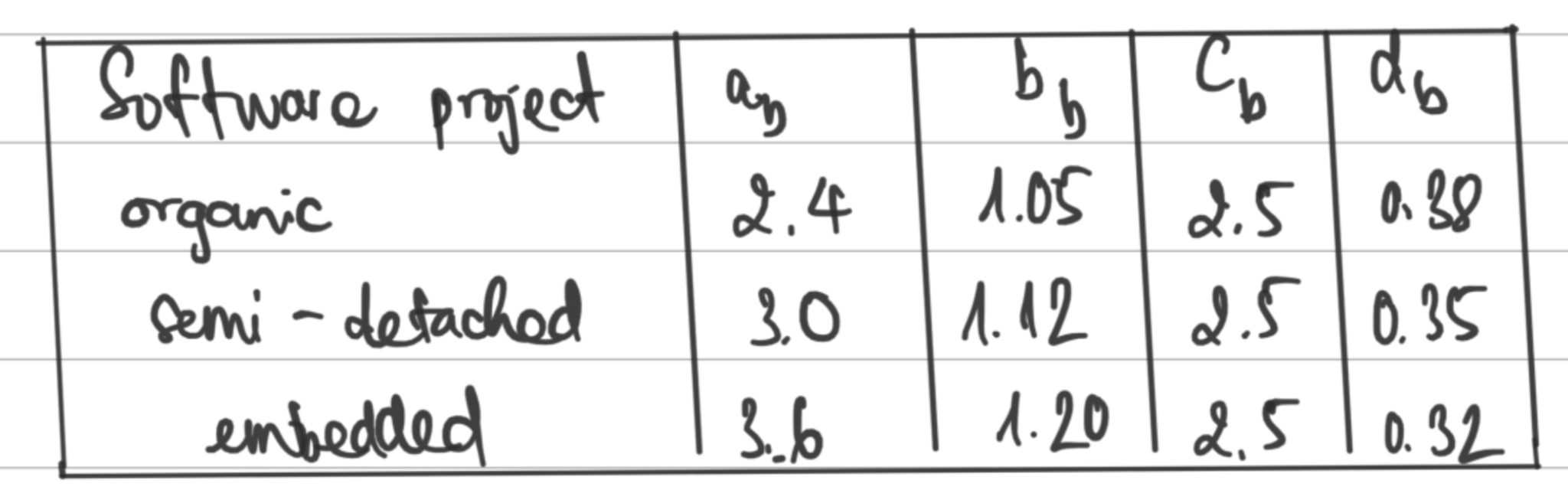
COCOMO có 3 loại: organic mode (quen thuộc); semi-detached mode (ở giữa), embedded mode (ràng buộc cứng nhắc)

=> Trong bài này, nhóm dự án sử dụng mô hình Basic Model / Basic COCOMO, với mô hình semi-detached mode.

# 2. Công thức tính



# 3. Quy ước



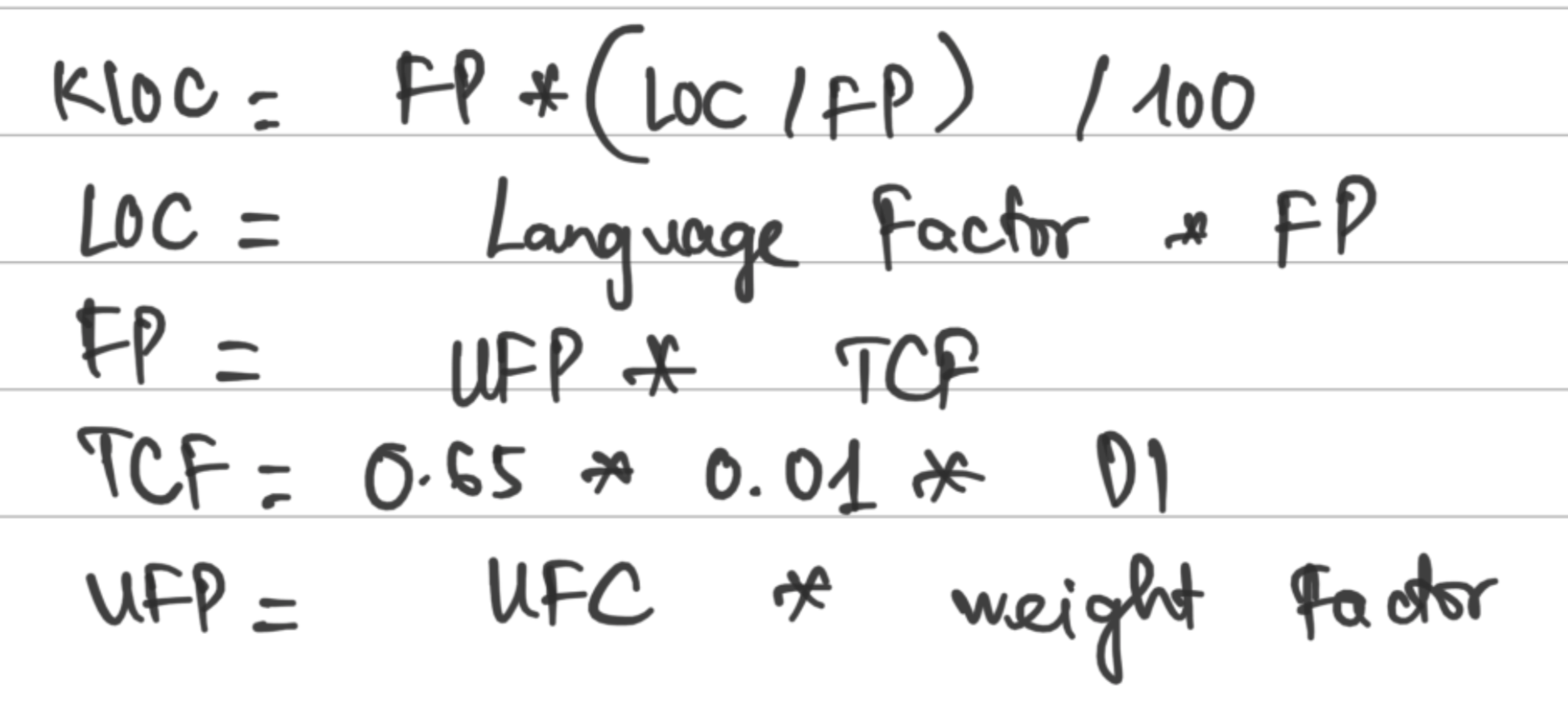
Theo những mục đã cung cấp ở trên, hiện tại, ta sẽ tính thời gian ước lượng theo mô hình Basic COCOMO, với công thức cần tính lần lượt là Effort Applied (E), Development Time (D) và People Required (P).

Để tính được 3 thông số trên, ta cần biết được KLOC (1k line of code).

Sau đây là những quy ước của các thuật ngữ trong bài này:

* Unadjusted Function Points: UFP
* Technical Complexity Factor: TCF
* Degree of Influence: DI
* Unadjusted Function Point Count: UFC

Công thức tính:



# 4. Tính toán

## Step 1: Tính số UFC

Ở bước này, ta sẽ đếm các danh mục cần có trong mỗi hạng mục yêu cầu. Cụ thể:

* External input:   
  External input trong phát triển phần mềm là dữ liệu đầu vào được cung cấp từ bên ngoài hệ thống, cụ thể ở đây khách hàng sẽ cung cấp gì cho team để xây dựng làm input đầu vào cho hệ thống  
  Trong bài toán này, External input gồm danh sách các yêu cầu cơ bản, chức năng mà khách hàng muốn có trong hệ thống mong muốn.  
  => Tổng số: 1
* External output  
  External output kết quả được trả về cho khách hàng. External output có thể là bất kỳ loại thông tin nào được gửi từ ứng dụng phần mềm đến người dùng hoặc các hệ thống khác, chẳng hạn như báo cáo, biểu đồ…  
  Trong bài toán này, External output gồm: danh sách các link quan trọng, agenda, gantt chart, yêu cầu chức năng, yêu cầu phi chức năng, product backlog, report team 1, design document, report team 2, UI UX, tài liệu thiết kế chi tiết, source code, tài liệu hướng dẫn quản trị (admin manuals), tài liệu hướng dẫn vận hành (operational manual), test case document, report test, tài liệu hướng dẫn người dùng (user manual).

=> Tổng số: 17

* External inquiry  
  External inquiry: là các dữ liệu đầu ra cần được kiểm chứng, cụ thể, chúng ta có thể kiểm chứng thông qua các đặc tả được cung cấp.

Trong bài toán này có tất cả 27 use cases.  
=> Tổng số: 27

* External files:   
  External files: là các file dùng để truyền thông tin tới các hệ thống khác ngoài hệ thống hiện tại sử dụng.

Trong bài toán này, ta có truyền thông tin sang một hệ thống ngoài đó là outlook (gửi email)

=> Tổng số 1.

* Internal files  
  Internal files: là các file logic chính trong hệ thống xử lý.

Trong bài toán này, chúng ta sử dụng 30 file sử dụng logic chính  
=> Tổng số 30.

## Step 2: Tính UFP

Giải thích về bảng thông tin trên. Bảng thông tin đưa trên là tổng hợp các danh mục ở step 1, được sắp xếp theo 3 mức độ hiện hành (simple, average, complex). Hiện tại trong bài toán này ta sử dụng mức độ simple để tránh phức tạp và hệ thống của chúng ta cũng không quá lớn về quy mô xử lý.

Ta có: External input (user input) có 1 file; External output (user output) có 17 files; External inquiry (user request) có 27 files; External files có 1 file; Internal files có 30 files.

Ta sẽ nhân lần lượt các thông số matching tương ứng nhau với thông số của cột simple.

Từ đó, ta có kết quả như sau:

UFP = (1\*3) + (17\*4) + (27\*3) + (1\*7) + (30\*5) = **309**

## Step 3: Technical Complexity Factor

Technical Complexity Factor (TCF) là một chỉ số trong phương pháp định lượng phần mềm COCOMO II để đo lường độ phức tạp kỹ thuật của phần mềm. TCF đánh giá các yếu tố kỹ thuật như khả năng tái sử dụng code, tầm quan trọng của hệ thống, khả năng mở rộng và bảo trì, giao diện người dùng, tích hợp với các ứng dụng khác, độ tin cậy và sự an toàn.

TCF gồm 9 chỉ số và tương ứng từng chỉ số, ta sẽ nhận định và đánh giá từng danh mục trên thang điểm 5.

1. Scalability: *The ability of a system to handle an increasing workload or number of users without sacrificing performance.* ***(2)***
2. Interoperability: *The ease with which a system can integrate with other systems or applications.* ***(2)***
3. Security: *The measures in place to protect a system from unauthorized access or attacks.* ***(3)***
4. Availability: *The extent to which a system is available and accessible to users.* ***(4)***
5. Performance: *The speed and efficiency with which a system performs its tasks.* ***(4)***
6. Maintainability: *The degree to which a system can be easily modified and updated over time.* ***(3)***
7. Reliability: *The ability of a system to perform consistently and predictably without error or failure.* ***(2)***
8. Complexity of algorithms: *The complexity of the algorithms used by the system to perform its tasks.* ***(3)***
9. Data storage requirements: *The amount and complexity of data that must be stored and managed by the system.* ***(2)***
10. Technical skills required: *The level of technical expertise required to develop and maintain the system.* ***(5)***

## Step 4: Tính DI

Degree of Influence là chỉ số đo lường mức độ ảnh hưởng mà một yếu tố có thể gây ra đến một rủi ro cụ thể. Degree of influence đánh giá mức độ ảnh hưởng của một nhân tố nào đó đối với việc xảy ra một rủi ro.

Để tính được chỉ số Degree of Influence (DI), ta cộng tất cả các danh mục ở step 3.

Ta có **DI = 30**

## Step 5: Tính TCF dựa trên DI

Technical Complexity Factor (TCF) là một chỉ số trong phương pháp định lượng phần mềm COCOMO II để đo lường độ phức tạp kỹ thuật của phần mềm. TCF đánh giá các yếu tố kỹ thuật như khả năng tái sử dụng code, tầm quan trọng của hệ thống, khả năng mở rộng và bảo trì, giao diện người dùng, tích hợp với các ứng dụng khác, độ tin cậy và sự an toàn.

**TCF = 0.65 \* 0.01 \* DI = 0.65 \* 0.01 \* 30 = 0.195**

## Step 6: Tính FP

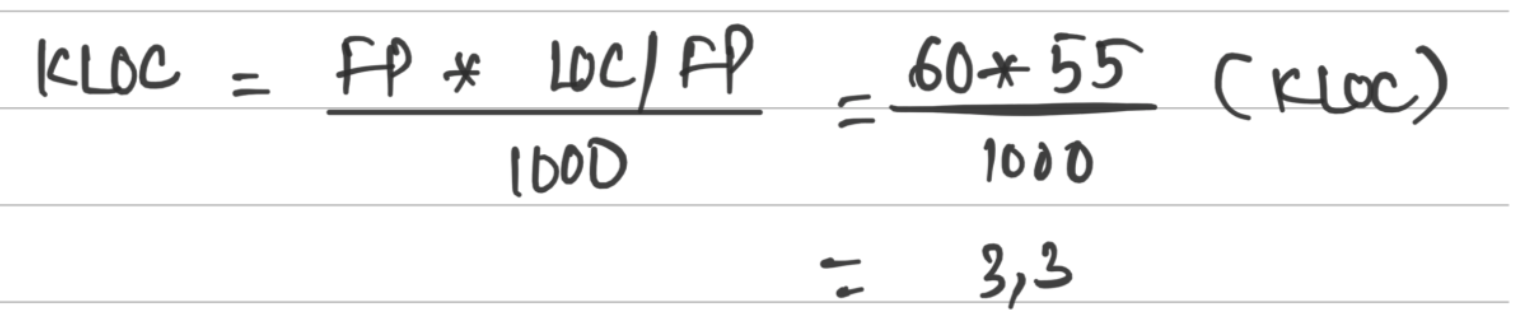
Function Point đánh giá kích thước của một hệ thống phần mềm bằng cách đếm và gán điểm cho các yêu cầu chức năng của hệ thống.

**FP = UFP \* TCF = 309 \* 0.195 = 60**

## Step 7: Tính KLOC

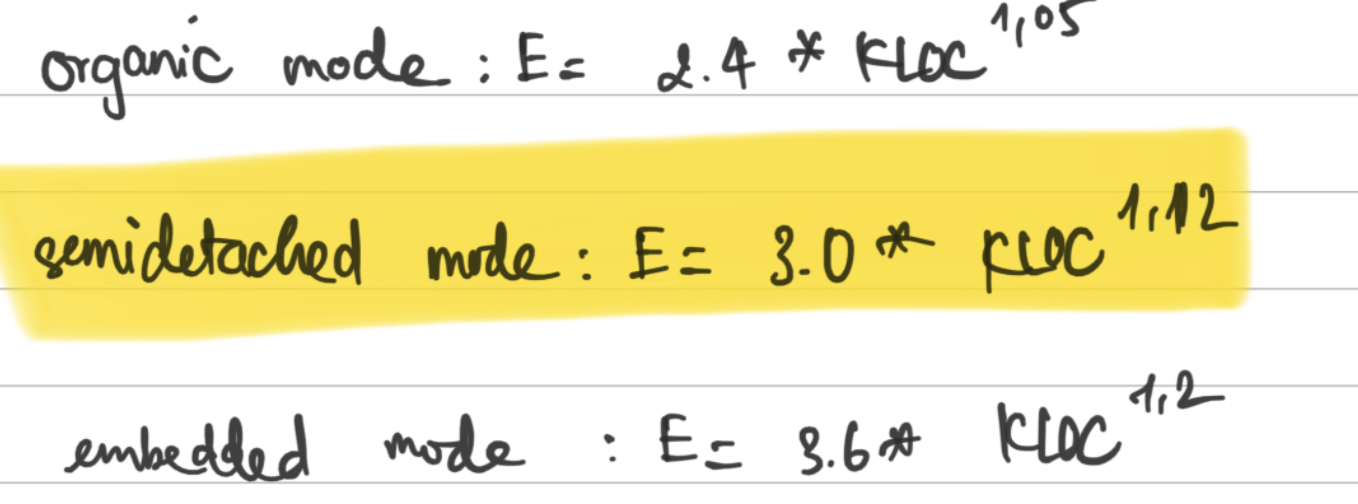
Cuối cùng, KLOC là viết tắt của Kilolines of Code, tức là đo lường kích thước của một chương trình máy tính bằng số lượng các dòng code. KLOC là một trong những phương pháp đo lường kích thước phần mềm phổ biến được sử dụng trong quản lý dự án phần mềm. KLOC được tính bằng cách đếm số lượng dòng code trong mã nguồn của chương trình máy tính. Để tính toán KLOC, mỗi dòng code chỉ được tính một lần, bao gồm cả các dòng chú thích và khoảng trống không cần thiết.  
  
 Dự án này được dự định viết bằng ngôn ngữ NodeJS, vì thế LOC/FP (theo google) là 53.

Vì thế, ta có kết quả KLOC như sau:



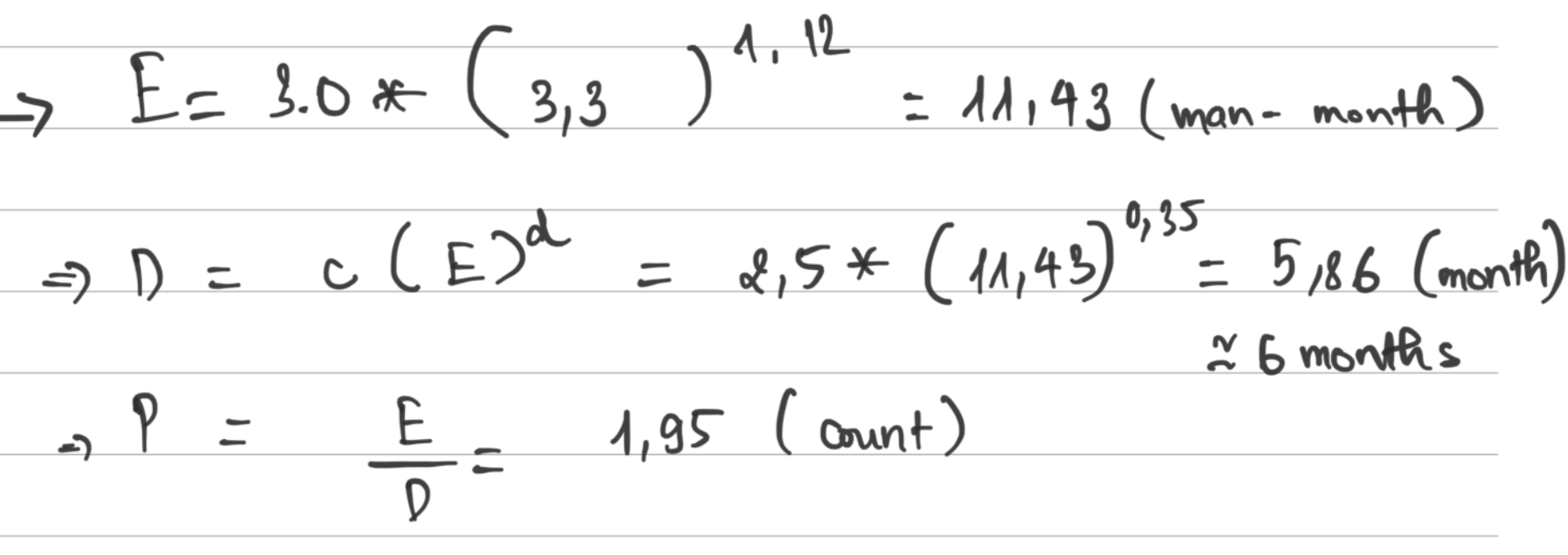
# 5. Kết luận

Ta có 3 phương pháp tính ra E tuỳ thuộc vào 3 loại mô hình.



Trong bài toán này, ta dùng mô hình semi-detached mode.

Suy ra, ta có các kết quả sau đây:



# 6. References

[Phat Trien DAPM - Google Drive](https://drive.google.com/drive/folders/1RaNIby59YulWL6wtJWpUz75VONhHEHZf)