

Môn học

**ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

---

**Chương IV**

**QUẢN LÝ CHẤT LƯỢNG**

# Nội dung

---

1. Kiểm soát tiến trình thực hiện dự án
2. Độ đo chất lượng
3. Chi phí chất lượng PM

# IV.1 Kiểm soát tiến trình thực hiện dự án

---

- (1) Kiểm soát các hoạt động quản lí rủi ro
- (2) Kiểm soát tiến độ dự án
- (3) Kiểm soát nguồn lực dự án
- (4) Kiểm soát ngân sách dự án

# (1) Kiểm soát các hoạt động quản lí rủi ro

---

- ❑ Các khoản mục rủi ro được xác định ở phần rà soát hợp đồng và lập kế hoạch dự án
- ❑ Các hoạt động quản lí rủi ro có hệ thống được sử dụng để ứng phó với các rủi ro.
- ❑ **Kiểm soát tiến trình quản lí rủi ro:**
  - Đánh giá định kỳ về tình trạng của các khoản mục rủi ro và kết quả mong đợi của các hoạt động quản lí rủi ro
  - Quản lí dự án sẽ can thiệp và giúp đưa ra giải pháp trong trường hợp cần thiết.

## (2) Kiểm soát tiến độ dự án

---

### ❑ Tại sao dự án bị chậm?

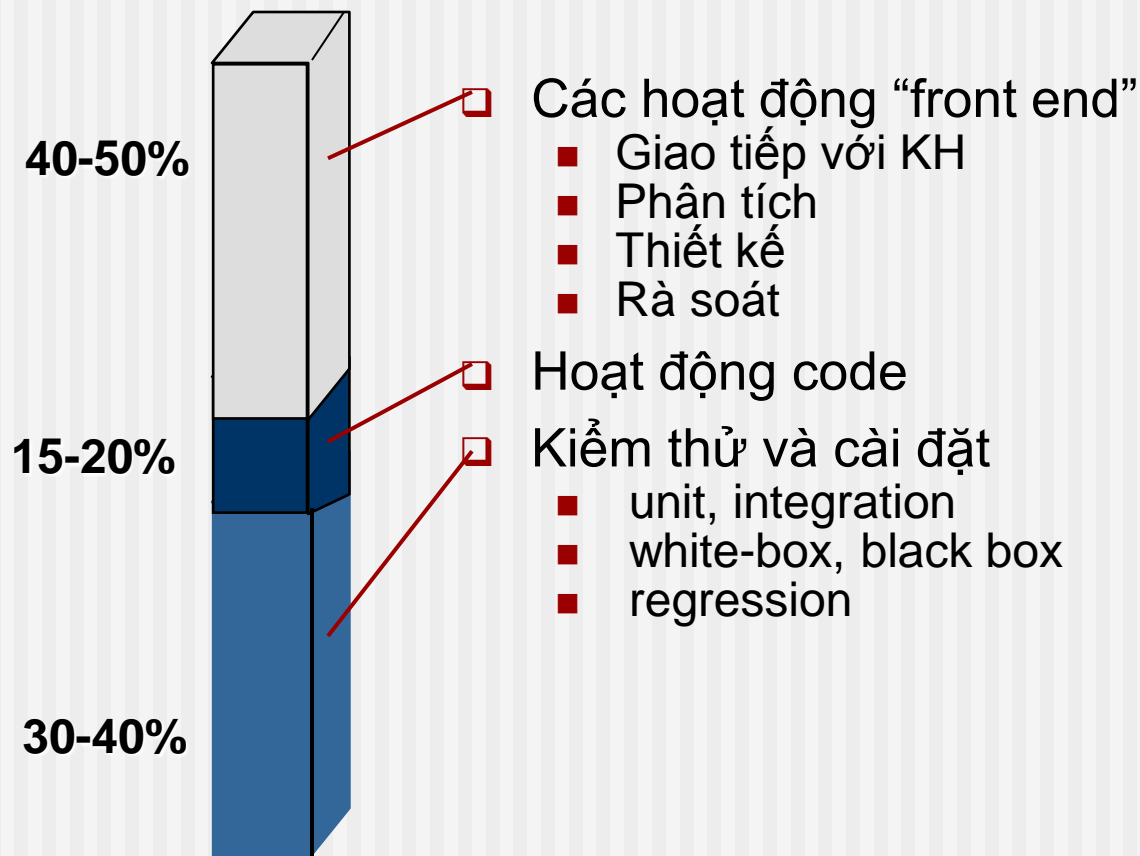
- Deadline không thực tế
- Thay đổi yêu cầu của KH mà không thay đổi lịch biểu.
- Đánh giá thấp về nỗ lực và nguồn lực để thực hiện công việc
- Các rủi ro có thể/không thể dự đoán đã không được xem xét khi dự án bắt đầu
- Khó khăn về kĩ thuật và con người không lường trước
- Sự nhầm lẫn trong giao tiếp giữa các nhân viên trong dự án

# Lập lịch biểu dự án

---

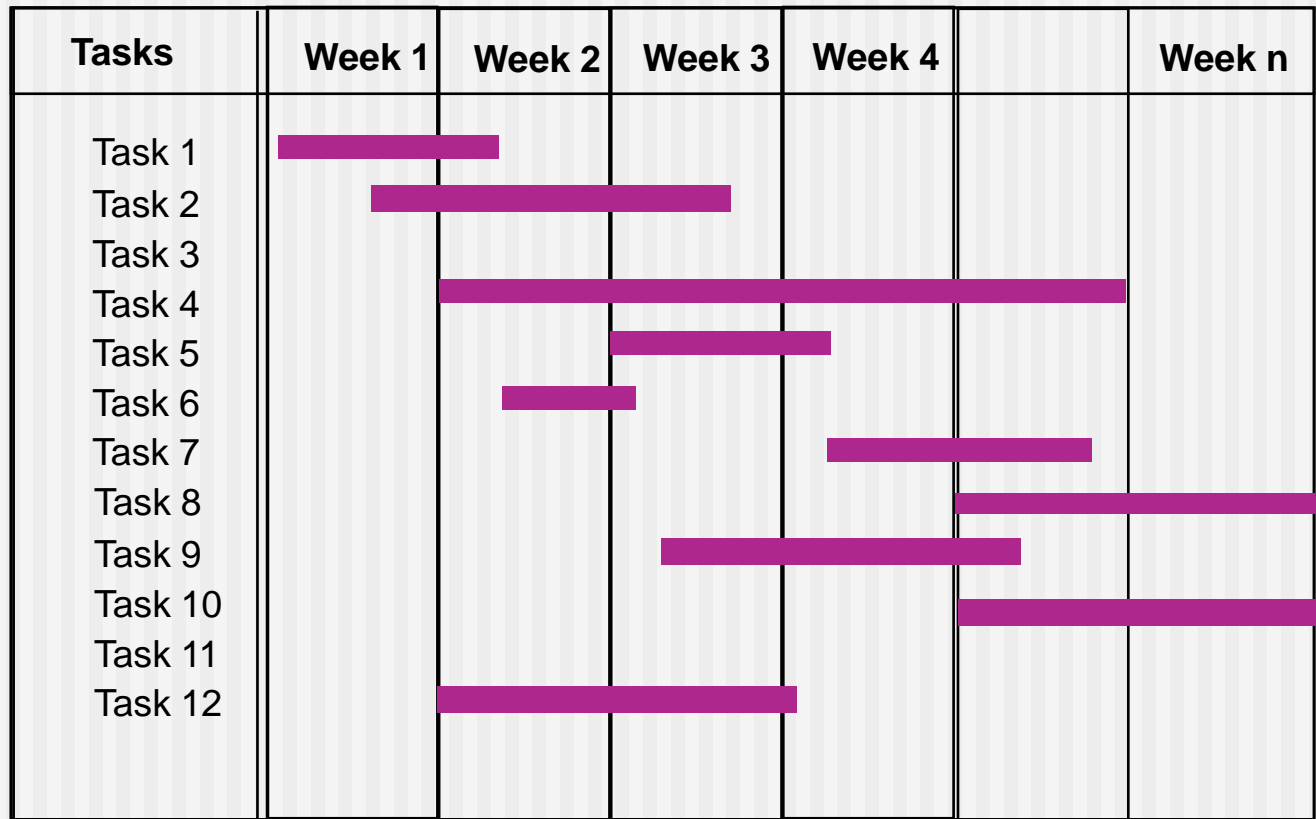
- ❑ Là hoạt động phân bổ thời gian cho các tác vụ và xác định thứ tự thực hiện chúng
- ❑ **Nguyên tắc lập lịch:**
  - Phân chia: xác định các tác vụ
  - Phụ thuộc lẫn nhau: xác định mối quan hệ giữa các tác vụ
  - Cấp phát thời gian
  - Đảm bảo nguồn lực thực hiện tác vụ luôn sẵn có
  - Xác định trách nhiệm
  - Xác định kết quả của mỗi tác vụ
  - Xác định các cột mốc của dự án

# Phân bổ nỗ lực thực hiện dự án



# Biểu đồ dòng thời gian (Time-line Chart)

---





# Công cụ để vẽ biểu đồ Time-line

---

- ☐ Microsoft Project
- ☐ Ganttic
- ☐ Gantt Project
- ☐ TeamGantt
- ☐ ....

# Theo dõi tiến độ (kiểm soát)

---

- ❑ Tổ chức các cuộc họp định kỳ để các thành viên của mỗi nhóm báo cáo về tiến độ thực hiện và các vấn đề gặp phải
- ❑ So sánh ngày bắt đầu thực tế với ngày bắt đầu dự kiến của mỗi tác vụ
- ❑ Dựa trên các cột mốc để nhận ra sự chậm trễ trong việc hoàn thành các tác vụ dự kiến
- ❑ Tập trung vào các chậm trễ nghiêm trọng gây ảnh hưởng đến deadline của dự án.
- ❑ Người quản lý can thiệp bằng cách:
  - Bổ sung nguồn lực
  - Đàm phán lại với khách hàng về tiến độ của dự án

# (3)Kiểm soát nguồn lực dự án

---

- ❑ Nguồn lực dự án gồm:
  - **Con người**
  - Môi trường phát triển PM
    - Công cụ phần mềm
    - Phần cứng
- ❑ Với mỗi nguồn lực, cần quản lí các thông tin:
  - Mô tả nguồn lực
  - Tính sẵn có
  - Thời gian tài nguyên được yêu cầu
  - Khoảng thời gian tài nguyên được sử dụng

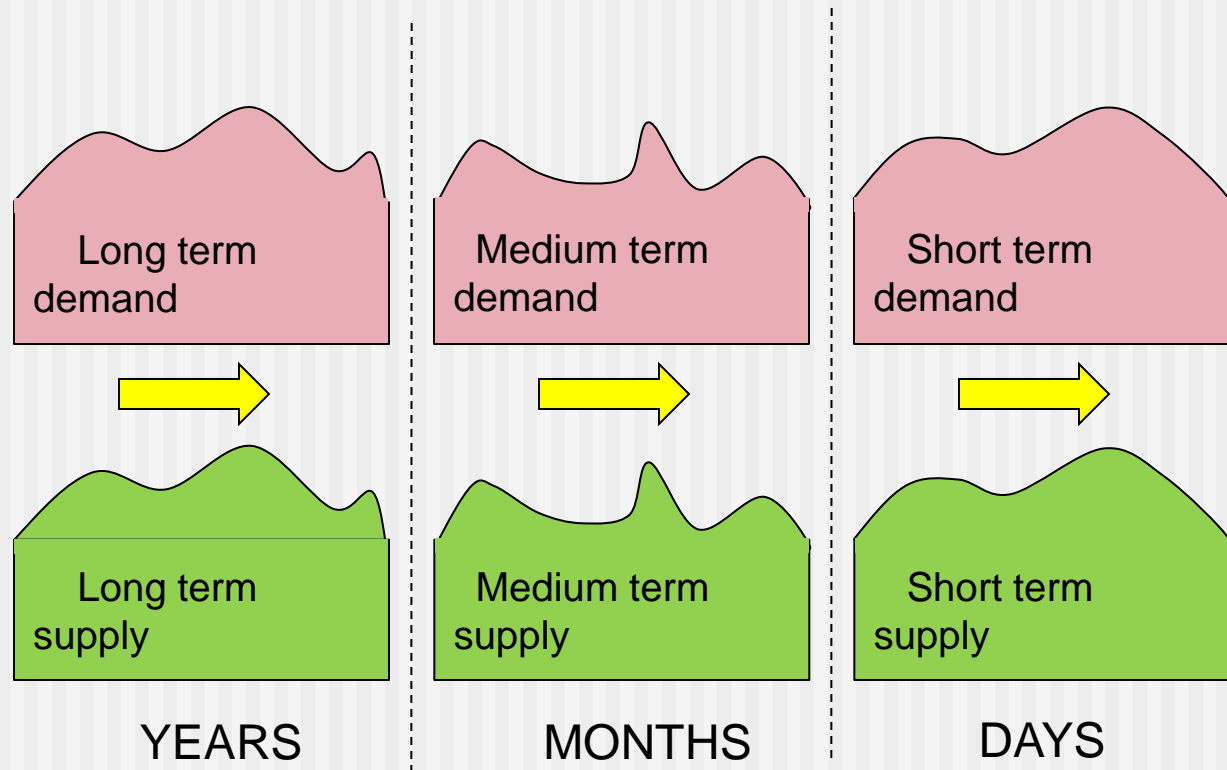
# Kiểm soát nguồn nhân lực (con người)

---

- ❑ Tối ưu việc phân bổ nguồn nhân lực cho các dự án
- ❑ Quản lí chi phí cho nguồn nhân lực
- ❑ Phát triển khả năng của nhân viên
- ❑ Phân công công việc phù hợp với năng lực của nhân viên

# Vạch kế hoạch sử dụng nguồn nhân lực

- ❑ Các yêu cầu về nguồn nhân lực có thể là dài hạn hoặc ngắn hạn



# Đánh giá tính sẵn có và năng lực

---

- ❑ Năng lực

- Kỹ năng, trình độ, kinh nghiệm, địa điểm,...

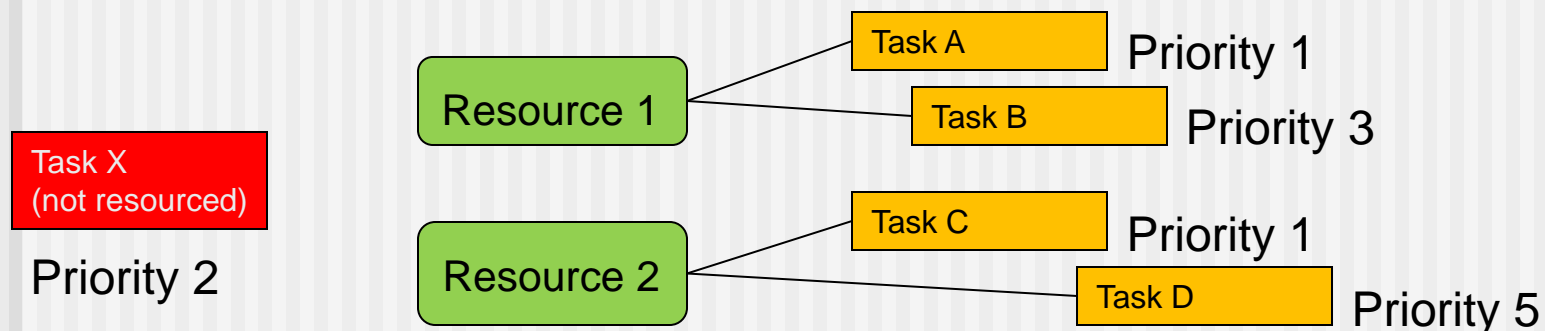
- ❑ Tính sẵn có

- Khoảng thời gian được yêu cầu cho công việc không bị gián đoạn bởi các hoạt động khác

# Đánh giá tính sẵn có và năng lực

❑ Năng lực hay tính sẵn có quan trọng hơn?

→ Người quản lý dự án nên tìm kiếm nguồn lực có khả năng nhất để thực hiện công việc



## (4) Kiểm soát ngân sách dự án

---

### ❑ Quản lí ngân sách

- Kiểm soát chi phí dự án trong phạm vi ngân sách đã được phê duyệt

### ❑ Các nhiệm vụ quản lí ngân sách

- Dự toán ngân sách
- Lập ngân sách dự án
- Kiểm soát ngân sách



# Budget baseline

---

- ❑ Khi ngân sách được rà soát và kiểm duyệt cho bước kế tiếp sẽ tạo ra Budget baseline
- ❑ Được sử dụng để kiểm soát ngân sách

# Kiểm soát ngân sách dự án

---

- ❑ Kiểm soát các thay đổi đối với ngân sách dự án
- ❑ So sánh chi phí thực tế với budget baseline → phát hiện sớm vấn đề vượt ngân sách dự kiến
- ❑ Giám sát thực hiện ngân sách
- ❑ Dựa trên các báo cáo ngân sách định kỳ

# Thực hiện ngân sách

---

- ☐ Xem xét chi phí dự án có đang được sử dụng theo kế hoạch?
- ☐ Xác định sai lệch
- ☐ Thực hiện hoạt động khắc phục

# Quản lí giá trị thu được (Earned Value Management)

---

- ❑ Quản lí giá trị thu được
  - Là kĩ thuật để đo lường tiến trình thực hiện của dự án
  - Cảnh báo sớm về những vấn đề thực hiện dự án
  - Kiểm soát chi phí và tiến độ dự án

# Quản lí giá trị thu được (Earned Value Management)

---

- ❑ Để thực hiện EVM, cần xác định 3 giá trị:
  - Giá trị dự kiến (PV)
  - Chi phí thực tế (AC)
  - Giá trị thu được (EV)

# Quản lí giá trị thu được (Earned Value Management)

---

## □ Giá trị dự kiến (PV)

- Tổng ngân sách cho các tác vụ được vạch kế hoạch *tại một thời điểm* trong lịch biểu
- $BCWS_i$  : là ngân sách cho tác vụ  $i$

$$PV = \sum (BCWS_i)$$

## □ Chi phí thực tế (AC)

- Tổng chi chí thực tế cho các tác vụ đã được hoàn thành *tại một thời điểm* trong lịch biểu

# Quản lí giá trị thu được (Earned Value Management)

---

## □ Giá trị thu được (EV)

- Tổng ngân sách cho các tác vụ đã thực sự hoàn thành *tại một thời điểm* trong lịch biểu

# Các chỉ số tiến trình

---

## ❑ Chênh lệch chi phí (CV)

- Chênh lệch giữa ngân sách (chi phí dự kiến) và chi phí thực tế

$$CV = EV - AC$$

## ❑ Chênh lệch chi phí do thay đổi tiến độ (SV)

- Chênh lệch giữa việc hoàn thành tác vụ đúng tiến độ và thực tế

$$SV = EV - PV$$



# Các chỉ số tiến trình

---

## ❑ Chỉ số chi phí thực hiện (CPI)

- Giúp đánh giá ngân sách dự án có đang được sử dụng theo kế hoạch
- Được sử dụng để ước tính chi phí hoàn thành dự án

$$\text{CPI} = \text{EV/AC}$$

## ❑ Chỉ số tiến độ thực hiện(SPI)

- Được sử dụng để ước tính thời gian hoàn thành dự án

$$\text{SPI} = \text{EV/PV}$$

# Ví dụ:

---

- ❑ Một dự án dự kiến hoàn thành trong 12 tháng. Tổng kinh phí cho dự án là 1,2 triệu đô. Dự án đang thực hiện đến tháng thứ 4 (33% thời gian đã được sử dụng). Nhưng thực tế, chỉ thực hiện được 25% các tác vụ và tiêu tốn 41% chi phí. Dự án có theo đúng kế hoạch không?

## IV.2 Độ đo chất lượng phần mềm

---

- ❑ Chất lượng phần mềm thường được thể hiện bằng các đặc trưng định tính: **dễ sử dụng, tin cậy, đúng đắn,...**
- ❑ Không thể đo trực tiếp các đặc trưng  
→ Cần các độ đo định lượng (đo gián tiếp)

# Độ đo

---

- ❑ Đo kích thước, hiệu quả, chất lượng thiết kế, độ tin cậy...
- ❑ Không tuyệt đối
- ❑ Cung cấp phương pháp đánh giá chất lượng một cách có hệ thống
- ❑ Cung cấp cái nhìn sâu sắc về chất lượng sản phẩm trong quá trình phát triển

# Độ đo hiệu quả

---

- ❑ Đơn giản, dễ tính toán
- ❑ Khách quan, nhất quán trong đơn vị tính
- ❑ Độc lập với ngôn ngữ lập trình
- ❑ Là cơ chế phản hồi chất lượng hiệu quả

# Phân loại độ đo

---

- (1) Độ đo sản phẩm (Product Metric)
- (2) Độ đo quy trình (Process Metric)

# (1) Độ đo sản phẩm

---

- ☐ Đo kích cỡ phần mềm
- ☐ Độ đo thiết kế
- ☐ Độ đo mã nguồn
- ☐ Độ đo kiểm thử
- ☐ Độ đo bảo trì

# Đo kích cỡ phần mềm

---

- ❑ Đo số dòng lệnh(LOC - lines of code)
- ❑ Đo điểm chức năng(FP - Function Point)



# Điểm chức năng(FP)

---

- ❑ **Điểm chức năng**: là đơn vị đo biểu diễn số chức năng mà hệ thống cung cấp cho người dùng.
- ❑ **Đo điểm chức năng**: đo tính năng của PM dựa trên yêu cầu người dùng

# Lợi ích của phân tích điểm chức năng (FPA)

---

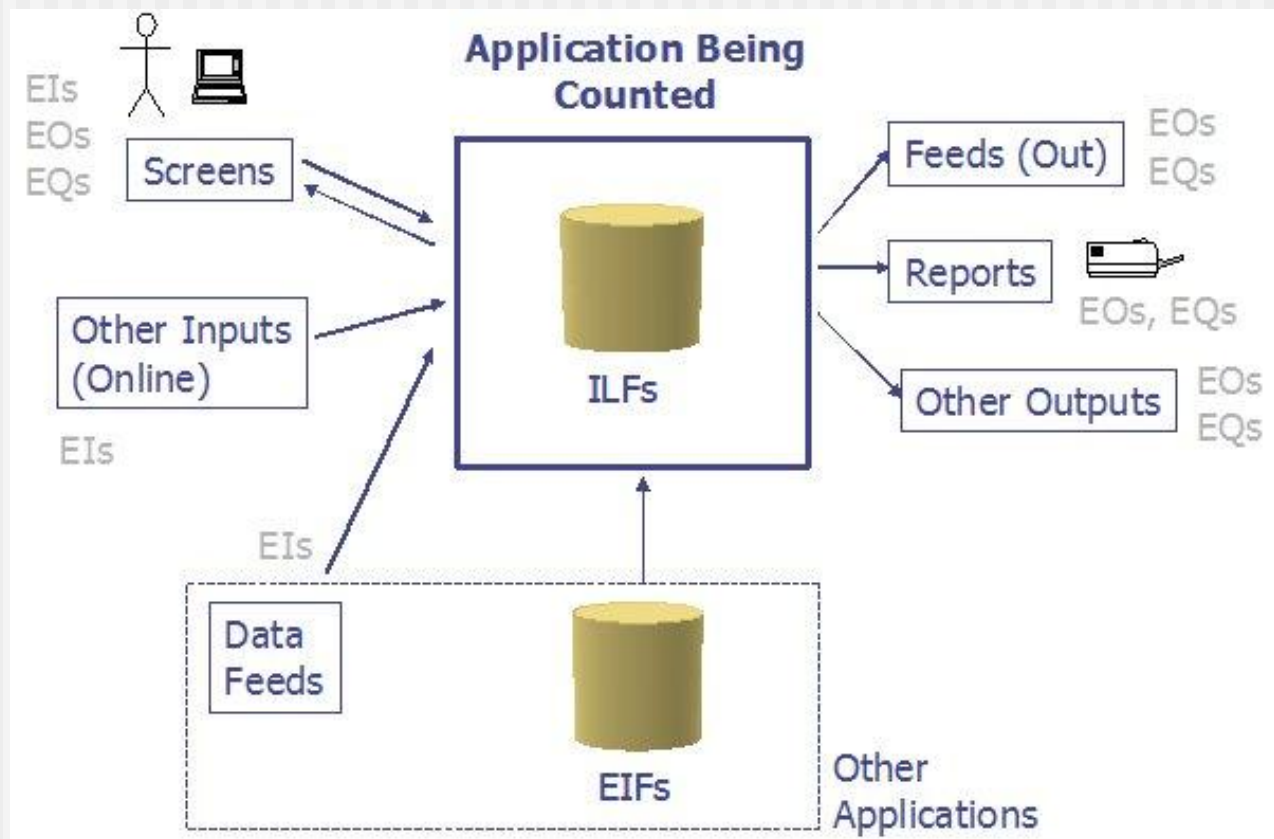
- Ứng dụng FPA như:
  - Công cụ xác định kích cỡ của PM
  - Công cụ giúp user xác định lợi ích của PM
  - Công cụ ước tính chi phí, tài nguyên được yêu cầu cho phát triển và bảo trì PM
  - Là nhân tố giúp so sánh PM

# Các bước thực hiện FPA

---

1. Đếm các FP chưa điều chỉnh (UFP)
2. Tính hệ số điều chỉnh (VAF)
3. Tính FP được điều chỉnh

# Bước 1: Đếm các FP chưa điều chỉnh (UFP)



# Bước 1: Đếm các FP chưa điều chỉnh

---

- ❑ Cần đếm 2 nhóm chức năng:
  - 1. **Data Functions(DF):**
    - 1. Internal logical files (ILF)
    - 2. External interface files (EIF)
  - 2. **Transactional Functions(TF):**
    - 1. External Inputs (EI)
    - 2. External Outputs (EO)
    - 3. External Inquiries (EQ)
- ❑ Mỗi chức năng được xếp hạng theo độ phức tạp: low, average, high

# Data Element Type(DET)

- ❑ Trường duy nhất (unique field)

The screenshot shows the Mendeley Research Networks Beta 0.9 sign-up page. The page has a header with the Mendeley logo and navigation links: 'Get Mendeley', 'How it works', and 'Blog'. The main content area contains a sign-up form with the following fields: First Name (Anton), Last Name (Litvinenko), E-mail (anton.li), Password, Confirm Password, and Research Field (Main discipline...). To the right of the form is a 'Sign in' section with e-mail and password fields, a 'Remember me' checkbox, and a 'forgot your password?' link. Below the sign-up form is a 'Sign up to:' section with a list of benefits: 'Import papers from other sites', 'Share resources with fellow researchers', 'Network to find new contacts', 'Synchronise your library across computers', and 'Manage your research library online'. At the bottom of this section, it says 'If you really don't want a Mendeley Web account you can still download Mendeley Desktop'. Green arrows point from the text 'DET-s' to various fields in the form, indicating that these fields are Data Element Types (DET-s).

**MENDELEY**  
RESEARCH NETWORKS BETA 0.9

**Get Mendeley** How it works Blog

First Name

Last Name

E-mail

Password

Confirm Password

Research Field

e-mail:  password:  **Sign in**

☐ Remember me [forgot your password?](#)

**DET-s**

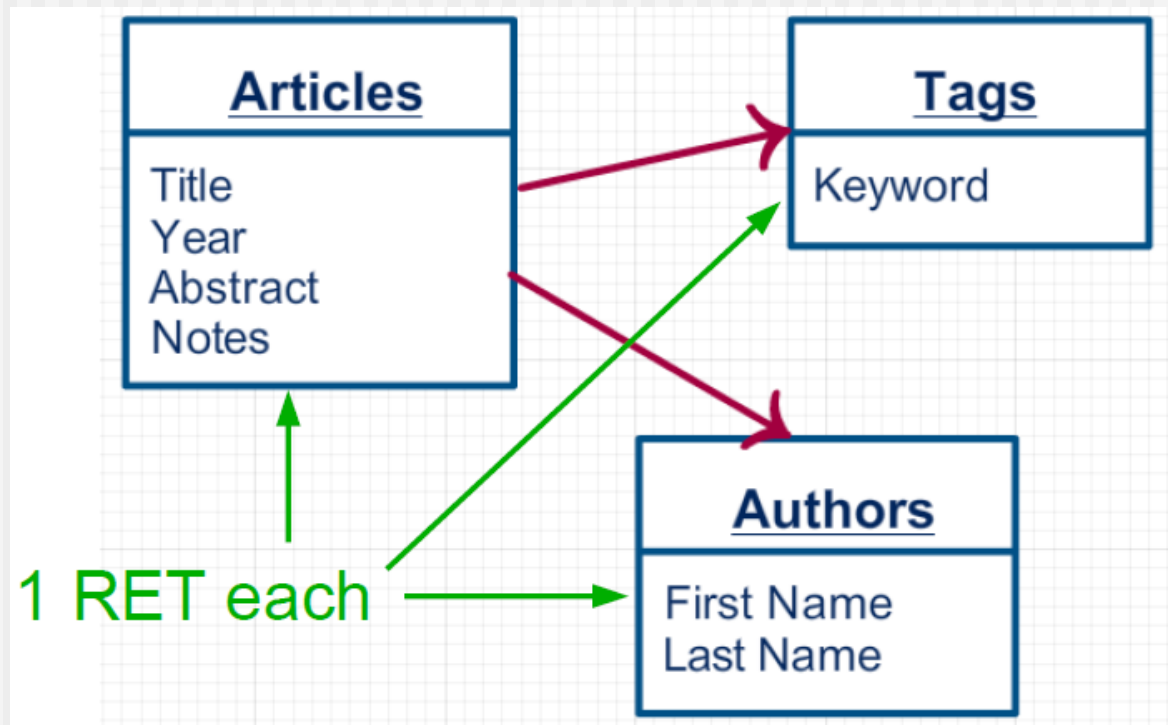
**Sign up to:**

- Import papers from other sites
- Share resources with fellow researchers
- Network to find new contacts
- Synchronise your library across computers
- Manage your research library online

If you really don't want a Mendeley Web account you can still download **Mendeley Desktop**

# Record Element Type (RET)

- ❑ Là nhóm các thành phần dữ liệu (Thường mỗi table là 1 RET)



# File Type Referenced (FTR)

- ❑ Ứng dụng tham chiếu tới thành phần (ILF, EIF)
- ❑ Ví dụ: soạn thảo bài viết
  - Abstract
  - Tags
  - Notes

## Abstract:

Enter the paper abstract here

## Tags:

Tag 1; Tag 2; ...

design; object-oriented; principles

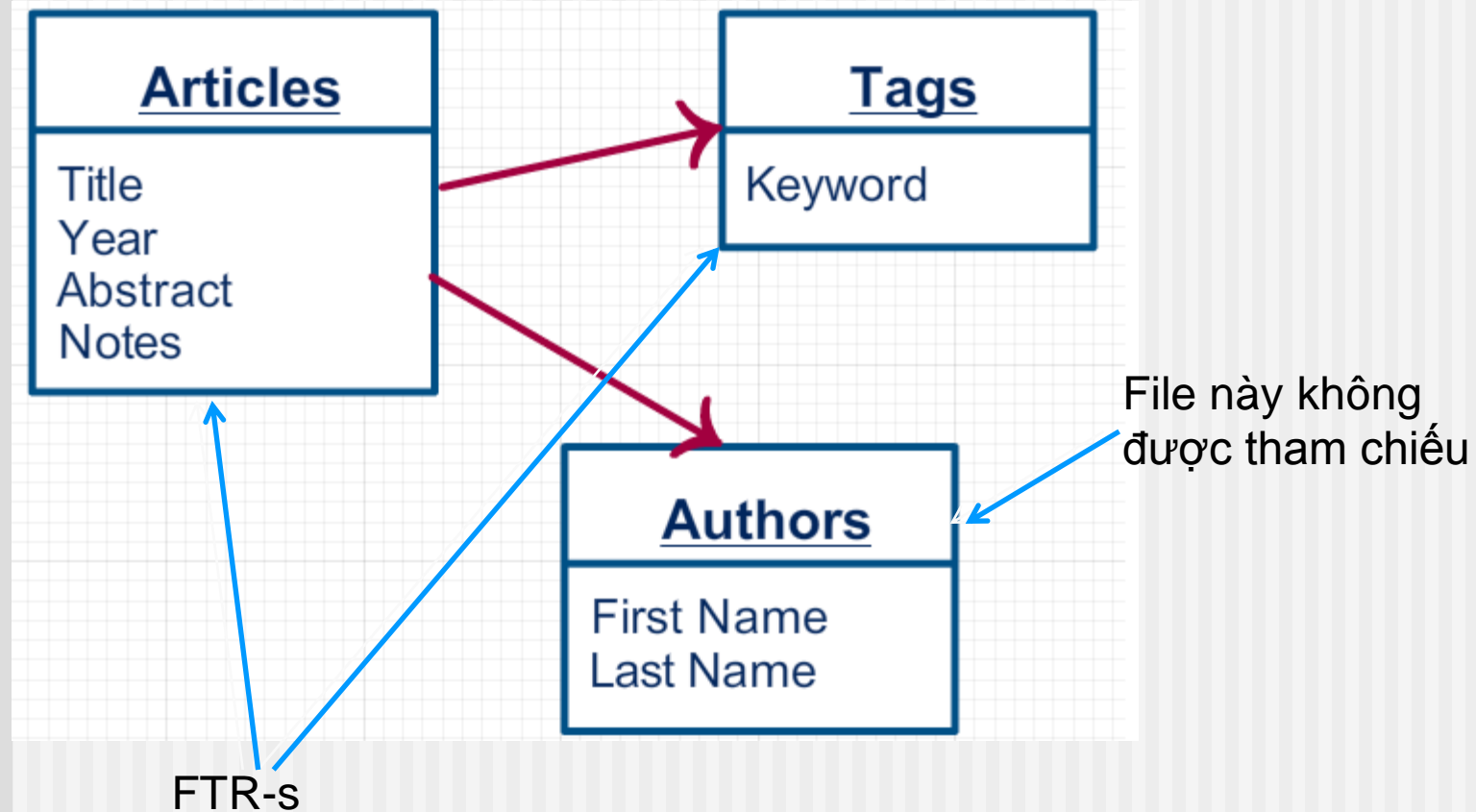
## Notes:

**B** *I* U   

[http://www.objectmentor.com/resources/articles/Principles\\_and\\_Patterns.pdf](http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf)



# File Type Referenced (FTR)



# File Type Referenced (FTR)

❑ Có bao nhiêu FTR?

★	●	📄	Authors	Title	Year	Published In	Added	▼	▲
☆	●	📄	Longstreet, D	Function Points Analysis Training Course	2005	Longstreet Consulting In...	Fri Sep 4 2009		
★	●	📄	Martin, RC	Design principles and design patterns	2000	Object Mentor	Sat Aug 29 2009		
☆	●	📄	Ford, Gary	Engineering Measurement for Software Engineers 5		Engineering	Fri Aug 28 2009		
☆	●	📄	Ford, Gary	Engineering Measurement for Software Engineers 4	1993	Engineering	Fri Aug 28 2009		
☆	●	📄	Ford, Gary	Engineering Measurement for Software Engineers 2		Engineering	Fri Aug 28 2009		
☆	●	📄	Ford, Gary	Engineering Measurement for Software	1993	Engineering	Fri Aug 28		

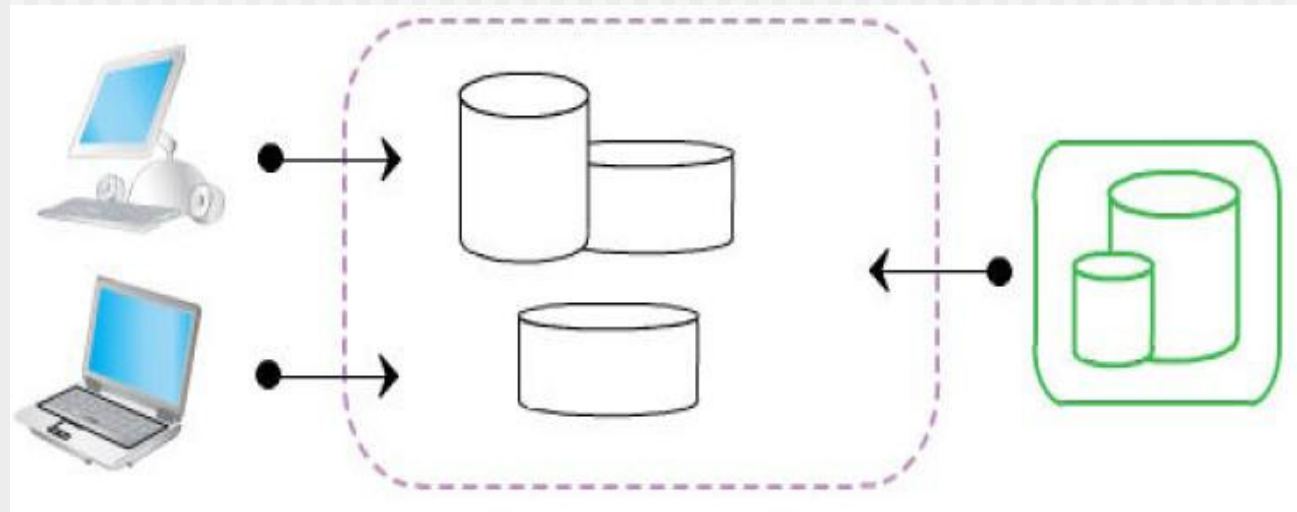
# Xếp hạng

---

Component	RET's	FTR's	DET's
External Inputs (EI)		✓	✓
External Outputs (EO)		✓	✓
External Inquiries (EQ)		✓	✓
External Interface Files (EIF)	✓		✓
Internal Logical Files (ILF)	✓		✓

# External Inputs

- ❑ Dữ liệu truyền vào ứng dụng: online, người dùng nhập, từ ứng dụng khác



# External Inputs

First Name

Last Name

E-mail

Password

Please fill in this field.

Confirm Password

Research Field

Please select a discipline.

Current Status

Please select your current status.

## Data Entry Preferences

- ☒ Pressing Enter moves between fields
- ☐ Automatically place decimal point
- ☒ Automatically recall last transaction for this name
- ☒ Warn when editing a transaction
- ☒ Warn when deleting a transaction or unused list item
- ☒ Beep when recording a transaction

Sort employee list by


☒ First Name ☐ Last Name

Time Format

Show portions of an hour as

☒ Decimal (10.20) ☐ Minutes (10:12)

 OK

 Cancel

 Help

Default

# External Inputs → Function Points

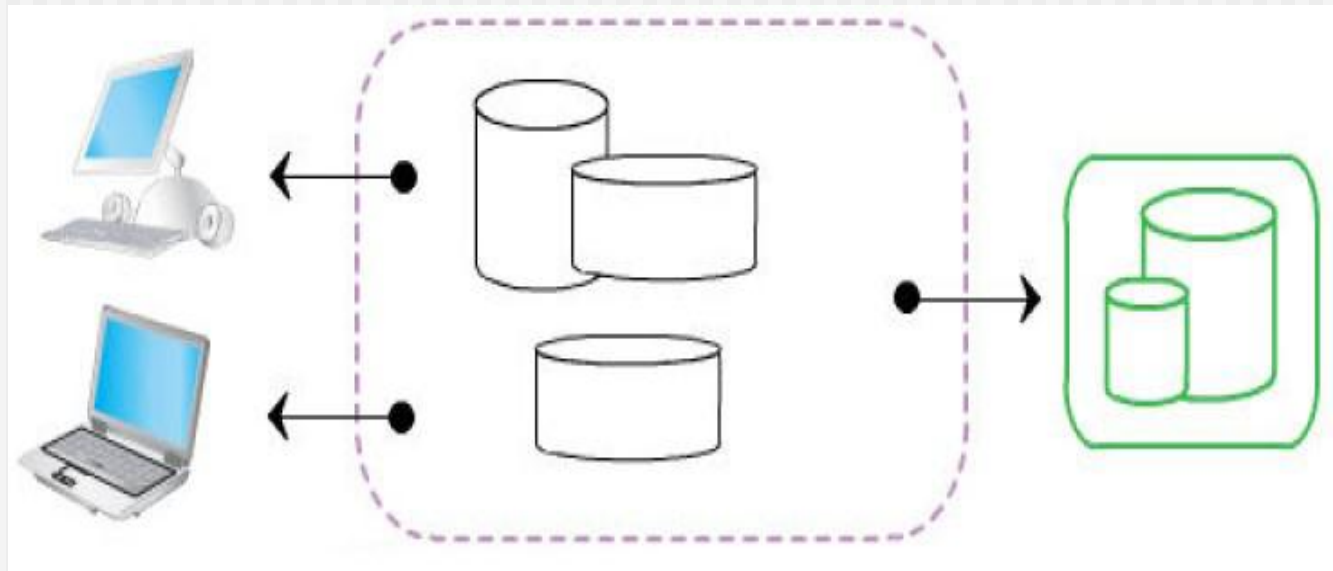
---

Files Type Referenced (FTR)	Data Elements (DET's)		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

- ❑ Low → 3 function points
- ❑ Average → 4 function points
- ❑ High → 6 function points

# External Outputs

- ❑ Dữ liệu xuất ra từ ứng dụng
  - Kết quả tính toán, báo cáo, đồ thị, biểu đồ



# External Outputs

Activity Level by Day of the Week				
	Day	Hits	% of Total Hits	User Sessions
1	Sun	1004	8.73%	111
2	Mon	1887	16.41%	201
3	Tue	1547	13.45%	177
4	Wed	1975	17.17%	195
5	Thu	1591	13.83%	191
6	Fri	2209	19.21%	200
7	Sat	1286	11.18%	121
	Total Weekdays	9209	80.08%	964
	Total Weekend	2290	19.91%	232

## Articles added and downloadable:

**3,989,644** added overall

**512,265** added in **Computer and Information Science**

**19,455** downloadable overall for free

**3,149** downloadable for free in **Computer and Information Science**



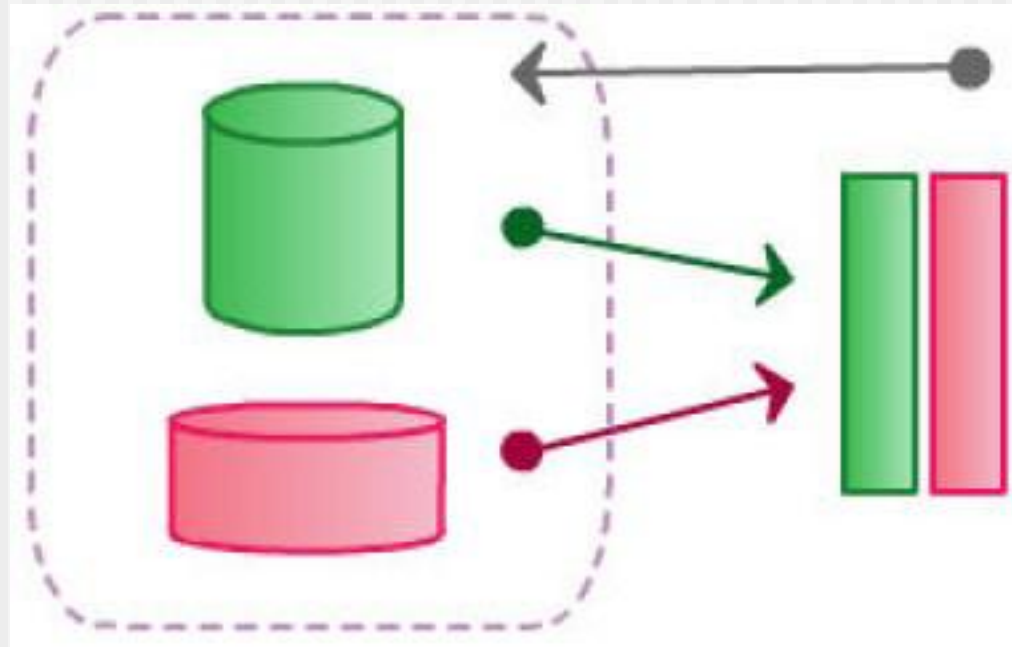
# External Outputs → Function Points

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

- ❑ Low → 4 function points
- ❑ Average → 5 function points
- ❑ High → 7 function points

# External Inquiries

- ❑ Dữ liệu lưu trữ + input, output



# External Inquiries

## State & County QuickFacts

Quick, easy access to facts about people, business, and geography

To begin, select a state from this list or use the map to the right.

Alabama



The population of the U.S. is 281,421,906 people (April 1, 2000). The population has grown 13.1% since 1990.

View more [USA QuickFacts](#).

See our [thematic maps](#)

Source: U.S. Census Bureau



# External Inquiries → Function Points

---

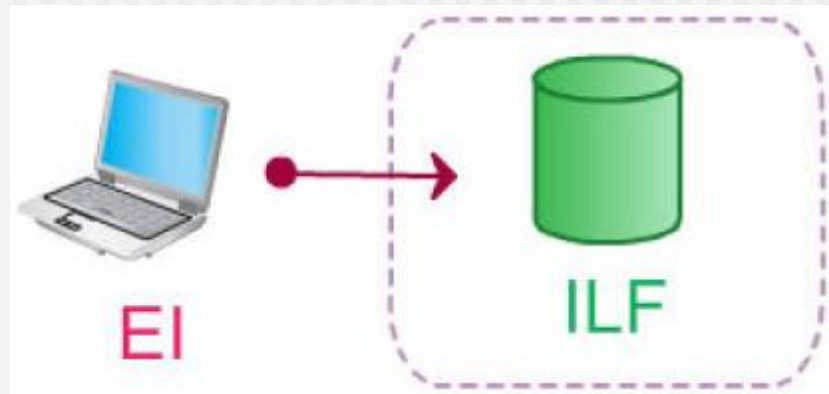
File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

- ❑ Low → 3 function points
- ❑ Average → 4 function points
- ❑ High → 6 function points

# Internal Logical Files(ILF)

---

- ❑ Dữ liệu nằm bên trong ranh giới của ứng dụng: dữ liệu nghiệp vụ, dữ liệu điều khiển
- ❑ Gồm:
  - RET (Table)
  - DET: các trường trong mỗi table



# Internal Logical Files → Function Point

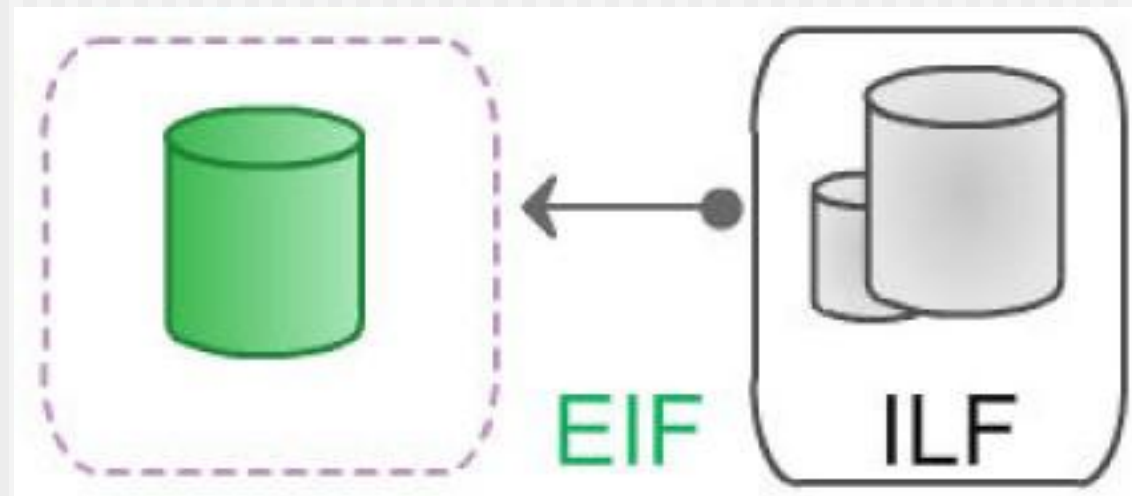
---

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)


- ❑ Low → 7 function points
- ❑ Average → 10 function points
- ❑ High → 15 function points

# External Interface Files(EIF)

- ❑ Dữ liệu bên ngoài ứng dụng, nhưng là dữ liệu nằm bên trong ứng dụng khác



# External Interface Files(EIF)

 These details need reviewing. You can mark them as correct, or search by title on Google Scholar.

Type:	<input type="text" value="Journal Article"/>
Title:	<input type="text" value="An introduction to game theory"/>
Authors:	<input type="text" value="Ricardson"/>
Journal:	<input type="text" value="Quality"/>
Volume:	<input type="text"/>
Issue:	<input type="text"/>
Pages:	<input type="text"/>
Year:	<input type="text" value="2003"/>



# External Interface Files → FP

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (5)	Low(5)	Average (7)
2 to 5 RET	Low (5)	Average (7)	High (10)
6 or More RET	Average (7)	High (10)	High (10)

- ❑ Low → 5 function points
- ❑ Average → 7 function points
- ❑ High → 10 function points

## Bước 2: Tính hệ số điều chỉnh(VAF)

- ☐ 0 = No Influence
- ☐ 1 = Incidental
- ☐ 2 = Moderate
- ☐ 3 = Average
- ☐ 4 = Significant
- ☐ 5 = Essential

$$VAF = 0.65 + \left[ \left( \sum_{i=1}^{14} C_i \right) / 100 \right]$$

General System Characteristics (GSCs)	Degree of Influence (DI) 0 - 5
1. Data Communications	_____
2. Distributed Data Processing	_____
3. Performance	_____
4. Heavily Used Configuration	_____
5. Transaction Rate	_____
6. Online Data Entry	_____
7. End-User Efficiency	_____
8. Online Update	_____
9. Complex Processing	_____
10. Reusability	_____
11. Installation Ease	_____
12. Operational Ease	_____
13. Multiple Sites	_____
14. Facilitate Change	_____
Total Degree of Influence (TDI)	_____
Value Adjustment Factor (VAF)	_____

$VAF = (TDI * 0.01) + 0.65$

# Bước 3: Tính FP được điều chỉnh

❑ *Tính theo công thức:*

$$FP = UFP * VAF$$

Type of Component	Complexity of Components			
	Low	Average	High	Total
External Inputs	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
External Outputs	___ x 4 = ___	___ x 5 = ___	___ x 7 = ___	
External Inquiries	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
Internal Logical Files	___ x 7 = ___	___ x 10 = ___	___ x 15 = ___	
External Interface Files	___ x 5 = ___	___ x 7 = ___	___ x 10 = ___	
Total Number of Unadjusted Function Points				_____

# Ưu và nhược điểm của FP

---

## ❑ Ưu điểm

- Có thể áp dụng để ước tính kích cỡ của PM ở những giai đoạn sớm.
- Dựa trên yêu cầu của người dùng, nên độ tin cậy của độ đo tương đối cao

## ❑ Nhược điểm

- Cần đặc tả yêu cầu thật chi tiết
- Nhiều yếu tố đánh giá mang tính chủ quan
- Cần đội ngũ có kinh nghiệm và nguồn lực đáng kể để tính
- Khó có thể áp dụng với tất cả các loại ứng dụng

# Độ đo thiết kế

---

## □ Độ phức tạp cấu trúc

- $S(i) = f_{\text{out}}^2(i)$
- $f_{\text{out}}(i)$  = số module cấp dưới trực tiếp của module  $i$

## □ Độ phức tạp dữ liệu

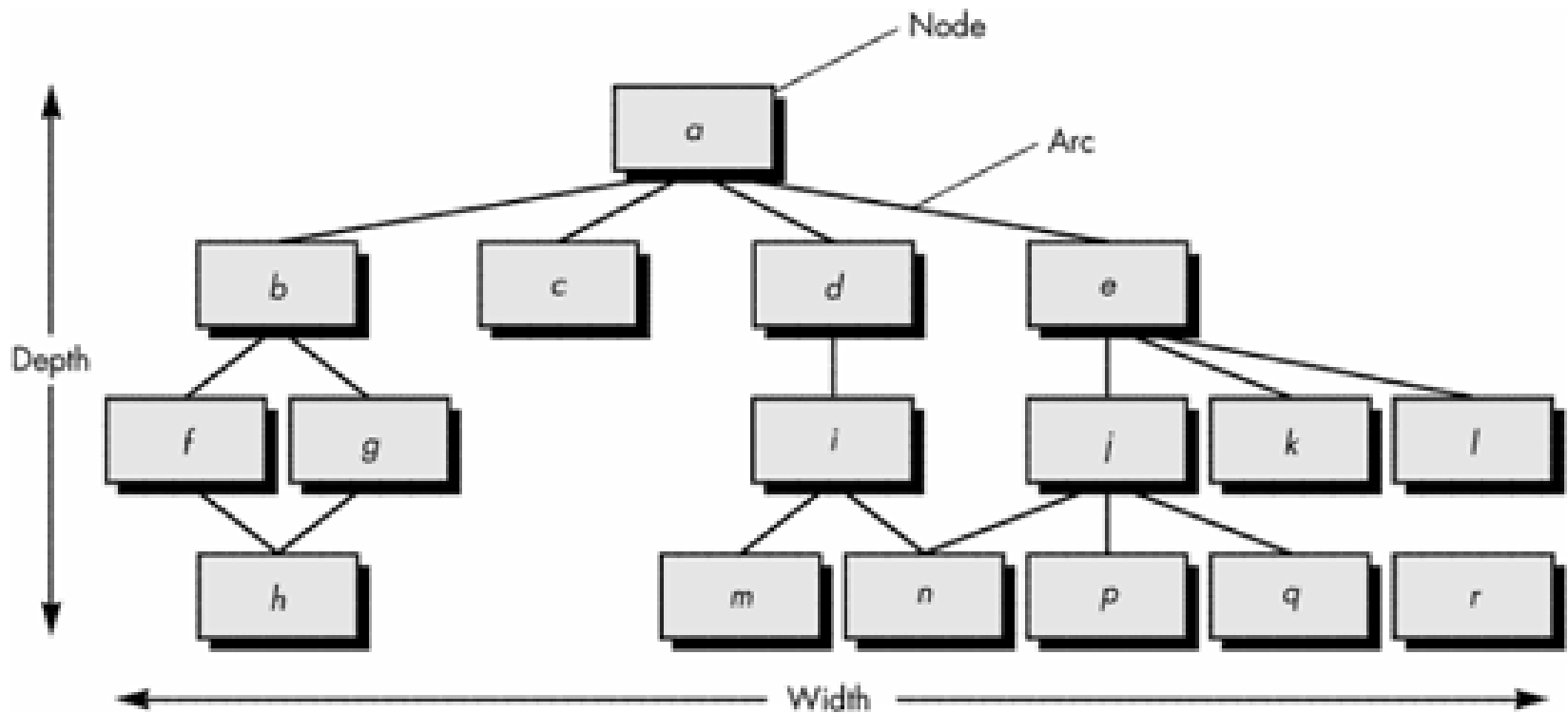
- $D(i) = v(i)/[f_{\text{out}}(i) + 1]$
- $v(i)$  = số inputs và outputs của module  $i$

## □ Độ phức tạp của hệ thống

$$C(i) = S(i) + D(i)$$

# Độ đo thiết kế mức cao(high level design)

---



# Độ đo thiết kế mức cao(high level design)

---

## □ Độ đo Morphology

- $\text{size} = n + a$
- $n$  : số modules(nút)
- $a$  : số đường điều khiển(cạnh)
- Depth: đường dài nhất từ gốc tới lá
- Width : số nút lớn nhất ở mức bất kì
- arc-to-node ratio:  $r = a/n$

# Độ đo thiết kế

---

- ❑ Chỉ số chất lượng cấu trúc thiết kế - DSQI  
(*Design structure quality index - IEEE Std. 982.1-1988*)
- ❑ Cấu trúc của PM được tạo nên từ
  - Chương trình
    - Module
    - Mối quan hệ giữa các module
  - Dữ liệu
    - Các file
    - Cấu trúc của file
    - Đặc trưng dữ liệu



# Chỉ số chất lượng cấu trúc thiết kế(DSQI)

---

## ❑ Bước 1: xác định các đại lượng

- **S1** : tổng số modules trong kiến trúc chương trình
- **S2** : số module phụ thuộc dữ liệu input hoặc tạo ra dữ liệu sẽ được sử dụng ở module khác
- **S3** : số module mà chức năng phụ thuộc vào xử lý trước đó
- **S4** : tổng số các khoản mục của cơ sở dữ liệu (gồm các đối tượng dữ liệu và tất cả các thuộc tính của nó)
- **S5** : số các khoản mục duy nhất (fields là unique)
- **S6** : số table trong cơ sở dữ liệu
- **S7** : số module có đầu vào và đầu ra duy nhất

# Chỉ số chất lượng cấu trúc thiết kế(DSQI)

---

## □ Bước 2: tính các giá trị trung gian

- $D_1 = 1$  nếu thiết kế được phát triển bằng phương pháp duy nhất, ngược lại  $= 0$
- $D_2 = 1 - (S2/S1)$  -- độ độc lập dữ liệu của module
- $D_3 = 1 - (S3/S1)$  – độ độc lập so với xử lý trước đó
- $D_4 = 1 - (S5/S4)$  – kích thước cơ sở dữ liệu
- $D_5 = 1 - (S6/S4)$  – độ phân chia cơ sở dữ liệu
- $D_6 = 1 - (S7/S1)$  – đặc trưng vào/ra của module

# Chỉ số chất lượng cấu trúc thiết kế(DSQI)

---

- ❑ Bước 3: tính chỉ số DSQI

$$DSQI = \sum w_i D_i$$

Trong đó,  $w_i$  – là trọng số thể hiện mức độ quan trọng của giá trị trung gian  $D_i$ ,  $i = 1, \dots, 6$ .

$$\sum w_i = 1$$

- ❑ Giá trị DSQI càng gần tới 1  $\rightarrow$  chất lượng càng cao.
- ❑ Là cơ sở để so sánh với các dự án trước đó
- ❑ Nếu DSQI quá thấp  $\rightarrow$  thiết kế lại.

# Độ đo mã nguồn(Halstead)

---

## □ Đo khối lượng chương trình

- $n_1$  = số các toán tử khác nhau
- $n_2$  = số các toán hạng khác nhau
- $N_1$  = tổng số lần xuất hiện của các toán tử
- $N_2$  = tổng số lần xuất hiện của các toán hạng

# Độ đo mã nguồn(Halstead)

---

- ❑ Độ dài:  $N = n_1 \log_2 n_1 + n_2 \log_2 n_2$
- ❑ Dung lượng:  $V = N \log_2 (n_1 + n_2)$
- ❑ Mức độ chương trình:  $L = 2n_2 / (n_1 N^2)$
- ❑ Độ khó của chương trình:  $D = 1/L$
- ❑ Nỗ lực kiểm thử:  $E = V/L$
- ❑ Mức độ ngôn ngữ:  $\lambda = L^2 V$

# Độ đo kiểm thử

---

- ❑ Độ bao phủ Test Case (TC):
  - $(\text{Số TC được thực thi} / \text{tổng số TC}) * 100$
- ❑ Hiệu quả của TC
  - Số TC tìm ra lỗi / tổng số TC
- ❑ Đánh giá nỗ lực kiểm thử
  - $(\text{Nỗ lực kiểm thử} / \text{tổng nỗ lực của dự án}) * 100$

# Độ đo bảo trì

---

## ❑ Software Maturity Index (SMI)

- $M_T$  = số module trong lần phát hành hiện tại
- $F_c$  = số module bị thay đổi trong lần phát hành hiện tại
- $F_a$  = số module được thêm vào trong lần phát hành hiện tại
- $F_d$  = số module bị bỏ đi trong lần phát hành hiện tại

# Độ đo bảo trì

---

- ❑ Software Maturity Index (SMI)

$$SMI = [M_T - (F_c + F_a + F_d)] / M_T$$

- ❑  $SMI \rightarrow 1$ , sản phẩm bắt đầu ổn định



## (2) Độ đo quy trình

---

- ❑ Tại sao cần đo quy trình?
  - Đánh giá tình trạng của một dự án đang triển khai
  - Theo dõi những rủi ro tiềm ẩn
  - Phát hiện ra các vấn đề trước khi chúng trở nên nghiêm trọng
  - Điều chỉnh luồng công việc
  - Đánh giá khả năng của nhóm dự án để kiểm soát chất lượng sản phẩm phần mềm

# Các yếu tố ảnh hưởng đến chất lượng phần mềm

---

- ❑ Kỹ năng của con người
- ❑ Độ phức tạp của sản phẩm
- ❑ Công nghệ
- ❑ Môi trường phát triển PM
- ❑ Thời gian hoàn thành sản phẩm
- ❑ Khách hàng

# Đo lường quy trình

---

## ❑ Dựa vào các kết quả của quy trình

- Các lỗi được phát hiện trước khi phát hành PM
- Các khiếm khuyết được báo cáo bởi người dùng cuối
- Sản phẩm công việc được giao
- Nguồn lực và thời gian đã tiêu tốn
- Sự đáp ứng tiến độ

# Đo lường quy trình

---

## □ Lưu ý:

- Không sử dụng độ đo để đánh giá hoặc đe dọa các cá nhân hoặc các nhóm
- Dữ liệu sau khi đo không nên xem là tiêu cực → được dùng để cải tiến quy trình
- Không nên chỉ dựa vào 1 độ đo đơn lẻ

# Các loại độ đo quy trình

---

- ❑ Đo chất lượng
  - Đo mật độ lỗi
- ❑ Đo lịch biểu
- ❑ Đo tính hiệu quả của loại bỏ lỗi
- ❑ Đo năng suất

# Đo mật độ lỗi

Code	Name	Calculation Formula
CED	Code Error Density	$CED = NCE/KLOC$
DED	Development Error Density	$DED = NDE/KLOC$

## ❖ Chú thích

- NCE – số lỗi được tìm thấy trong code khi rà soát và kiểm thử code
- KLOC – số dòng code
- NDE – tổng số lỗi được tìm thấy trong quá trình phát triển phần mềm

# Đo lịch biểu

Code	Name	Calculation Formula
TTO	Time Table Observance	$TTO = MSOT/MS$
ADMC	Average Delay of Milestone Completion	$ADMC = TCDAM/MS$

## ❖ Chú thích

- MSOT– các cột mốc được hoàn thành đúng thời gian
- MS – tổng số cột mốc
- TCDAM – tổng thời gian chậm trễ (ngày, tuần,..) của tất cả cột mốc

# Đo tính hiệu quả của loại bỏ lỗi

Code	Name	Calculation Formula
DERE	Development Errors Removal Effectiveness	$DERE = NDE / (NDE + NYF)$

## ❖ Chú thích

- NDE – số lỗi được tìm thấy trong quá trình phát triển phần mềm (Trước khi chuyển giao cho người dùng)
- NYF – số lỗi được tìm thấy trong giai đoạn bảo trì (sau khi chuyển giao)



# Đo năng suất

Code	Name	Calculation Formula
DevP	Development Productivity	$DevP = DevH/KLOC$
FDevP	Function point Development Productivity	$FDevP = DevH/NFP$

## ❖ Chú thích

- DevH– Tổng số giờ làm việc đầu tư cho phát triển hệ thống phần mềm

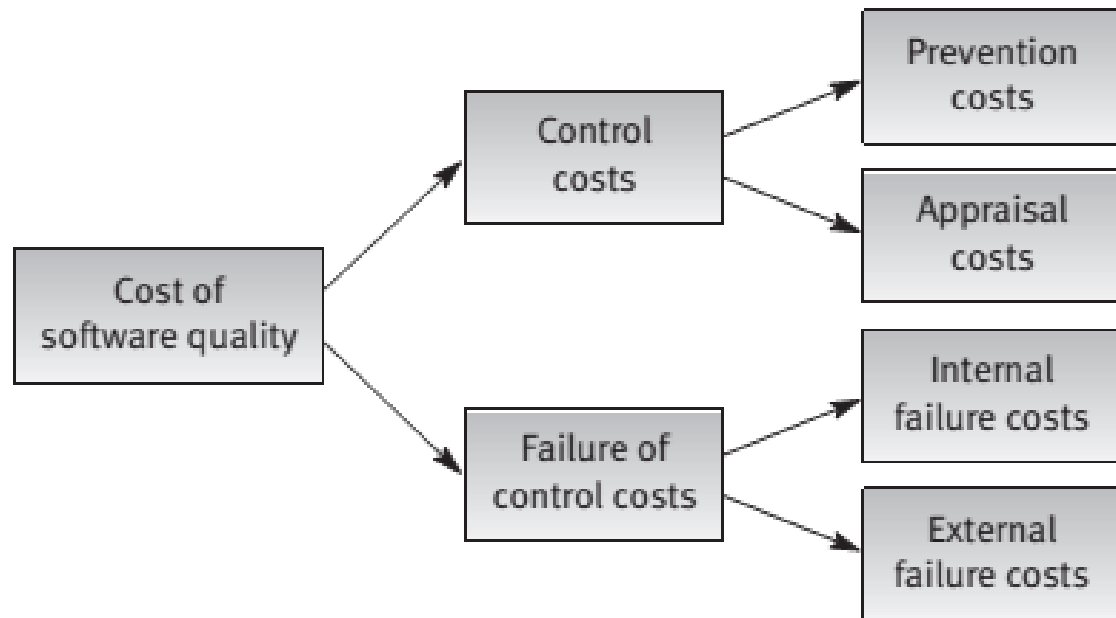
## IV.3 Chi phí chất lượng PM

---

- ❑ Mô hình chi phí chất lượng
- ❑ Cân bằng chi phí chất lượng

# Mô hình chi phí chất lượng

---



# Chi phí phòng ngừa (Prevention Costs)

---

- ❑ **Đầu tư phát triển cơ sở hạ tầng SQA**
  - Thủ tục và chỉ dẫn
  - Templates và checklists
  - Hệ thống quản lý cấu hình phần mềm
  - Độ đo chất lượng phần mềm
- ❑ **Thực hiện các hoạt động phòng ngừa:**
  - Huấn luyện nhân viên mới về SQA
  - Chứng nhận nhân viên
  - Tư vấn về các vấn đề SQA cho các leaders

# Chi phí phòng ngừa (Prevention Costs)

---

- ❑ **Kiểm soát hệ thống SQA thông qua việc thực hiện:**
  - Rà soát chất lượng nội bộ
  - Kiểm toán chất lượng bên ngoài
  - Rà soát chất lượng quản lý

# Chi phí phát hiện lỗi (Appraisal Costs)

---

- ❑ Chi phí rà soát
- ❑ Chi phí kiểm thử

# Chi phí thất bại nội bộ (Internal Failure Costs)

---

- ❑ Chi phí phát sinh để sửa các lỗi được phát hiện trước khi PM được phát hành
  - Chi phí thiết kế lại hoặc sửa chữa thiết kế sau rà soát và kiểm thử
  - Chi phí lập trình lại hoặc sửa chữa chương trình sau kiểm thử
  - Chi phí cho rà soát lại và kiểm thử hồi quy sau khi sửa các lỗi

# Chi phí thất bại bên ngoài (External Failure Costs)

---

- ❑ Chi phí phát sinh sau khi PM đã đưa vào vận hành
  - Giải quyết khiếu nại của khách hàng trong thời gian bảo trì.
  - Sửa lỗi được tìm thấy trong giai đoạn vận hành
  - Các tổn thất phải trả cho khách hàng trong trường hợp thất bại PM gây hậu quả nghiêm trọng.
  - ...



# Cân bằng chi phí chất lượng

