

BÀI TẬP THỰC HÀNH SỐ 2

THỰC HÀNH ƯỚC LƯỢNG KÍCH THƯỚC VÀ CHI PHÍ PHẦN MỀM

1. MỤC TIÊU

Sau khi kết thúc thực hành, các nhóm phải hoàn thành các nhiệm vụ sau:

- Ước lượng kích thước phần mềm theo phương pháp Functional Point (FP)
- Ước lượng chi phí phần mềm theo phương pháp COCOMO2

2. CƠ SỞ LÝ THUYẾT

2.1. Ước lượng kích thước phần mềm theo phương pháp FP

2.1.1. Quy trình ước lượng

Quy trình ước lượng kích thước phần mềm theo phương pháp FP bao gồm 7 bước

1. Xác định kiểu của mỗi Determine the type of function point count.
2. Xác định phạm vi ước lượng và biên của ứng dụng
3. Xác định tất cả các chức năng dữ liệu (internal logical files và external interface files) và mức độ phức tạp của chúng.
4. Xác định tất cả các chức năng xử lý (external inputs, external outputs, and external inquiries) và mức độ phức tạp của chúng.
5. Xác định tổng giá trị FP chưa hiệu chỉnh.
6. Xác định giá trị của các nhân tố hiệu chỉnh dựa trên 14 đặc tính chung của hệ thống.
7. Tính toán giá trị FP đã hiệu chỉnh

2.1.2. Phân loại yêu cầu chức năng

Function Point (FP) là đơn vị đo lường số lượng chức năng nghiệp vụ mà một hệ thống thông tin (sản phẩm phần mềm) cung cấp cho người dùng. FP được dùng để đo lường kích thước phần mềm.

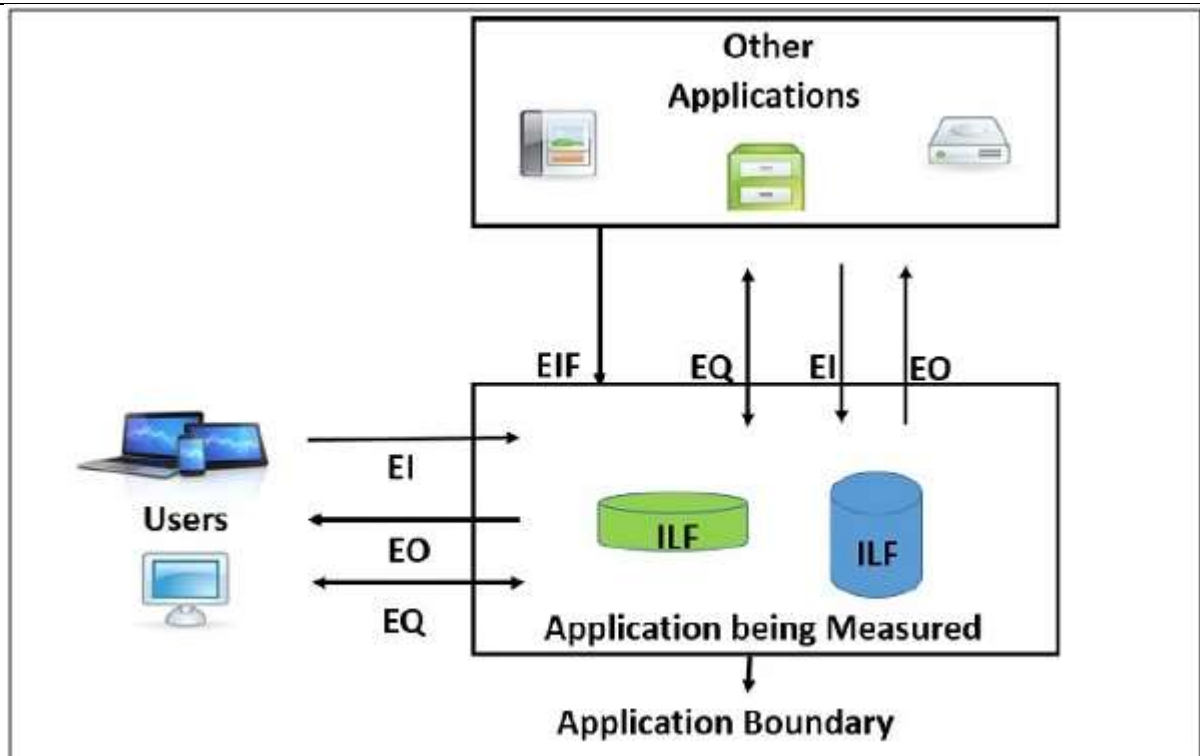
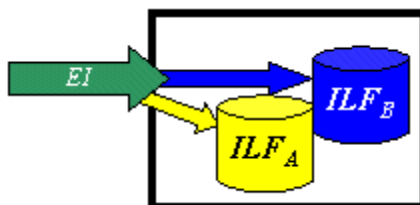


Figure 1: Application Boundary, Data Functions, Transaction Functions

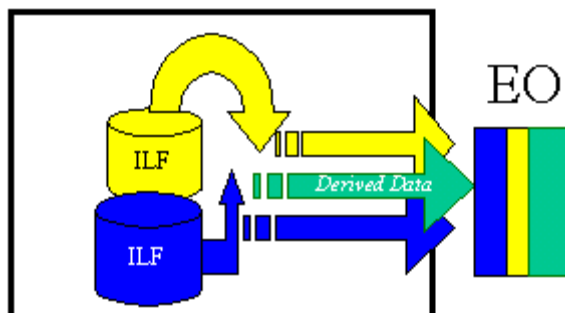
Các chức năng được chia làm 2 loại:

- **Chức năng dữ liệu (Data Functions):** có 02 loại
 - **Internal Logical Files (ILF):** số lượng các file logic nội tại
 - Là nhóm dữ liệu dạng logic nằm bên trong ứng dụng và được bảo trì thông qua các đầu vào bên ngoài hệ thống.
 - Là các bản ghi dữ liệu được sinh ra và được sử dụng, duy trì bởi hệ thống (thông qua các bảng biểu CSDL, file XML, file lưu trữ trung gian mà từ đó dữ liệu được truy xuất hoặc ghi vào sau khi kết thúc thao tác xử lý)
 - Các dữ liệu này được lưu trữ hoàn toàn bên trong hệ thống đang xây dựng, và được truyền vào từ Dữ liệu đầu vào (EI)
 - **External Interface Files (EIF):** số lượng các file giao tiếp ngoài
 - Là nhóm dữ liệu logic nằm bên ngoài ứng dụng, các ứng dụng khác có thể sử dụng dữ liệu này
 - Dữ liệu này được lưu trữ hoàn toàn bên ngoài ứng dụng đang xây dựng, được lưu trữ bởi các ứng dụng khác, và được dùng duy nhất cho mục đích tham khảo.
 - EIF của ứng dụng này sẽ là ILF của ứng dụng khác.
- **Chức năng nghiệp vụ xử lý (Transaction Functions):** có 03 loại
 - **External Inputs (EI):** số lượng các dữ liệu đầu vào từ bên ngoài hệ thống (qua thiết bị nhập). Dữ liệu đầu vào này có thể:
 - đến từ màn hình nhập liệu hoặc từ 1 ứng dụng khác
 - được lưu trữ trong một hoặc nhiều Internal Logical Files
 - có thể là thông tin điều khiển hoặc thông tin nghiệp vụ (nếu dữ liệu là thông tin điều khiển, nó phải không cập nhật một tập tin logic bên trong (ILF))
 - Hình dưới đây thể hiện một EI đơn giản cập nhật 2 ILF



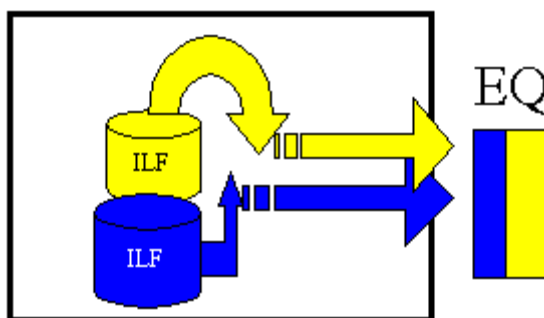
○ **External Outputs (EO):** số lượng các dữ liệu đầu ra bên ngoài hệ thống

- Thường được lấy từ bên trong hệ thống và cung cấp thông tin cho người sử dụng hoặc gửi tới ứng dụng khác (báo biểu, màn hình, thông báo lỗi, ...)
- Hình dưới đây thể hiện một Dữ liệu đầu ra (EO) đang kết xuất dữ liệu từ 2 file ILF



○ **External Inquiries (EQ):** số lượng các câu truy vấn dữ liệu ngoài

- Truy vấn dữ liệu và kết xuất ra giao diện ngoài
- Bao gồm cả truy vấn dữ liệu từ file logic nội tại (ILF) và file giao tiếp ngoài (EIF).
- Các dữ liệu truy vấn không được cập nhật vào các file logic nội tại (ILF) và cũng không kết xuất ra các file giao tiếp ngoài (EIF).
- Hình dưới đây thể hiện một EQ đang truy vấn 2 ILF:



Lưu ý: Nếu bạn lấy dữ liệu từ 1 ứng dụng và thêm nó vào 1 file trong ứng dụng của bạn, đây là thao tác kết hợp giữa GET (external inquiry) và ADD (external input)

2.1.3. Phân loại mức độ phức tạp của yêu cầu chức năng

Mỗi yếu tố sẽ được đánh giá theo 03 mức: Low, Average và High.

Sau khi các chức năng đã được phân chia thành 1 trong 5 loại (EI, EO, EQ, ILF và EIF), và được gán các giá trị Low, Average hoặc High.

- Với các chức năng xử lý (EI, EO, EQ), việc phân hạng sẽ dựa trên số lượng file được cập nhật hoặc tham chiếu đến (FTR – File Type Referenced), và số lượng kiểu thành phần dữ liệu (DET – Data Element Type).
- Với các chức năng dữ liệu (ILF, EIF), việc phân hạng sẽ dựa trên các kiểu bản ghi thành phần (RET – Record Element Type), và số lượng kiểu thành phần dữ liệu (DET – Data Element Type).

Phân loại mức độ phức tạp cho từng loại chức năng xử lý

For ILF and EIF				For EO and EQ				For EI			
Record Element s	Data Elements			File Types	Data Elements			File Types	Data Elements		
	1 - 19	20 - 50	51+		1 - 5	6 - 19	20+		1 - 4	5 - 15	16+
1	Low	Low	Avg	0 or 1	Low	Low	Avg	0 or 1	Low	Low	Avg
2 - 5	Low	Avg	High	2 - 3	Low	Avg	High	2 - 3	Low	Avg	High
6+	Avg	High	High	4+	Avg	High	High	3+	Avg	High	High

Function Type	Complexity-Weight		
	Low	Average	High
Internal Logical Files	7	10	15
External Interfaces Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

2.1.4. Nhân tố hiệu chỉnh

Nhân tố hiệu chỉnh VAF được tính dựa trên 14 đặc tính chung của hệ thống, và được đánh giá theo các mức độ từ 0 (không ảnh hưởng) đến 5 (ảnh hưởng trầm trọng).

Đặc tính chung của hệ thống		Mô tả tóm tắt
1.	Data communications	Có nhiều phương thức giao tiếp để trao đổi thông tin với ứng dụng/hệ thống không?
2.	Distributed data processing	Dữ liệu phân bố và xử lý như thế nào?
3.	Performance	Yêu cầu của người dùng về thời gian trả lời (năng suất) của hệ thống?
4.	Heavily used configuration	Hệ thống phần cứng hiện tại mà ứng dụng sẽ triển khai có khó sử dụng hay không?
5.	Transaction rate	Tần suất sử dụng của nghiệp vụ xử lý: hàng ngày, hàng tuần, hàng tháng, ...?
6.	On-Line data entry	Bao nhiêu % thông tin được nhập trực tuyến?
7.	End-user efficiency	Ứng dụng được thiết kế cho người dùng cuối có hiệu quả không?
8.	On-Line update	Có nhiều ILF được cập nhật bởi các thao tác xử lý trực tuyến không?
9.	Complex processing	Ứng dụng có đòi hỏi xử lý toán học hoặc xử lý logic phức tạp không?

10.	Reusability	Ứng dụng được phát triển để đáp ứng nhu cầu của 1 hay nhiều người sử dụng?
11.	Installation ease	Cài đặt ứng dụng có khó không?
12.	Operational ease	Các thủ tục khởi tạo, backup và phục hồi có được tự động hóa hiệu quả không?
13.	Multiple sites	Ứng dụng có được thiết kế, phát triển và hỗ trợ để cài đặt tại nhiều địa điểm cho nhiều tổ chức hay không?
14.	Facilitate change	Ứng dụng có được thiết kế, phát triển và hỗ trợ để dễ dàng thay đổi không?

2.1.5. Tính tổng số FP

Tổng số FP sẽ được tính theo công thức như bảng dưới đây:

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				
Multiplied Value Adjustment Factor				
Total Adjusted Function Points				

2.1.6. Biểu mẫu

Sử dụng biểu mẫu: Function Point Counting Template

Tham khảo file ví dụ: Function Point Counting Sample

2.2. Ước lượng chi phí phần mềm theo phương pháp COCOMO2

Sau khi đã ước lượng được số lượng FP chưa hiệu chỉnh (từ phần trên), hãy sử dụng trang sau để tính toán:

<http://csse.usc.edu/tools/cocomoi.php>.

Tùy theo giai đoạn đang ước lượng, các em hãy hiệu chỉnh các tham số đi kèm. Dưới đây là hướng dẫn và chú giải về các tham số trong từng mô hình

2.2.1. Các mô hình COCOMO 2

- Mô hình **Application Composition**: mô hình này giả định rằng hệ thống được tạo nên từ các thành phần có thể tái sử dụng. Mô hình này được thiết kế để ước lượng công sức phát triển bản mẫu (prototype).
- Mô hình **Early design**: được sử dụng tại giai đoạn thiết kế kiến trúc khi các yêu cầu đã có sẵn (và thiết kế chi tiết vẫn chưa bắt đầu).
- Mô hình **Reuse**: được sử dụng để tính công sức tích hợp các thành phần có thể sử dụng lại và/hoặc mã chương trình được sinh tự động bởi các công cụ dịch hay thiết kế chương trình. Nó thường được sử dụng kết hợp với mô hình Post-architecture.

- Mô hình **Post-architecture**: được sử dụng khi kiến trúc hệ thống đã được thiết kế và các thông tin chi tiết hơn về hệ thống đã có sẵn.

2.2.2. COCOMO 2 – Mô hình Early Design và Post-Architecture

Table I-1: Scale Factors for COCOMO II Early Design and Post-Architecture Models

Scale Factors (<i>W_i</i>)	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
RESL ³	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
PMAT	Weighted average of “Yes” answers to CMM Maturity Questionnaire					

Feature	Very Low	Nominal / High	Extra High
Precedentedness			
Organizational understanding of product objectives	General	Considerable	Thorough
Experience in working with related software Systems	Moderate	Considerable	Extensive
Concurrent development of associated new hardware and operational procedures	Extensive	Moderate	Some
Need for innovative data processing architectures, algorithms	Considerable	Some	Minimal
Development Flexibility			
Need for software conformance with preestablished requirements	Full	Considerable	Basic
Need for software conformance with external interface specifications	Full	Considerable	Basic
Premium on early completion	High	Medium	Low

Table I-3: RESL Rating Components

Characteristic	Very Low	Low	Nominal	High	Very High	Extra High
Risk Management Plan identifies all critical risk items, establishes milestones for resolving them by PDR.	None	Little	Some	Generally	Mostly	Fully
Schedule, budget, and internal milestones through PDR compatible with Risk Management Plan	None	Little	Some	Generally	Mostly	Fully
Percent of development schedule devoted to establishing architecture, given general product objectives	5	10	17	25	33	40
Percent of required top software architects available to project	20	40	60	80	100	120
Tool support available for resolving risk items, developing and verifying architectural specs	None	Little	Some	Good	Strong	Full
Level of uncertainty in Key architecture drivers: mission, user interface, COTS, hardware, technology, performance.	Extreme	Significant	Considerable	Some	Little	Very Little
Number and criticality of risk items	> 10 Critical	5-10 Critical	2-4 Critical	1 Critical	> 5Non-Critical	< 5 Non-Critical

Table I-4 : TEAM Rating Components

Characteristic	Very Low	Low	Nominal	High	Very High	Extra High
Consistency of stakeholder objectives and cultures	Little	Some	Basic	Considerable	Strong	Full
Ability, willingness of stakeholders to accommodate other stakeholders' objectives	Little	Some	Basic	Considerable	Strong	Full
Experience of stakeholders in operating as a team	None	Little	Little	Basic	Considerable	Extensive
Stakeholder teambuilding to achieve shared vision and commitments	None	Little	Little	Basic	Considerable	Extensive

Key Process Areas	Almost Always (>90%)	Often (60-90%)	About Half (40-60%)	Occasion- ally (10-40%)	Rarely If Ever (<10%)	Does Not Apply	Don't Know
Requirements Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Project Planning	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Project Tracking and Oversight	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Subcontract Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Quality Assurance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Configuration Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Process Focus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization Process Definition	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Training Program	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrated Software Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Product Engineering	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Intergroup Coordination	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Peer Reviews	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quantitative Process Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Software Quality Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defect Prevention	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technology Change Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Process Change Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Check Almost Always when the goals are consistently achieved and are well established in standard operating procedures (over 90% of the time).
- Check Frequently when the goals are achieved relatively often, but sometimes are omitted under difficult circumstances (about 60 to 90% of the time).
- Check About Half when the goals are achieved about half of the time (about 40 to 60% of the time).
- Check Occasionally when the goals are sometimes achieved, but less often (about 10 to 40% of the time).
- Check Rarely If Ever when the goals are rarely if ever achieved (less than 10% of the time).
- Check Does Not Apply when you have the required knowledge about your project or organization and the KPA, but you feel the KPA does not apply to your circumstances.
- Check Don't Know when you are uncertain about how to respond for the KPA.