

비교대상	단계 1: 응용 합성 (프로토타이핑)	단계 2: 초기 설계	단계 3: 설계 이후
크기	응용 포인트	기능 포인트 (FP)와 언어 종류	FP와 언어, LOC
재사용 effort	모델에 포함됨	ESLOC(Equiv. SLOC) 로 표현 ← 재사용 관련 변수 들의 함수로 계산. 관련변수들: SU(프로그램 이해에 어려움 정도), UNFM(프로그래머 친밀성), AA(프로그램 테스트 및 평가 노력) 등	ESLOC로 표현 ← 재사용 관련 변수 들의 함수로 계산
요구변경	모델에 포함됨	변경 비율이 비용승수로 반영됨	변경 비율이 비용 승수로 반영됨

비교대상	단계 1: 응용 합성 (프로토타이핑)	단계 2: 초기 설계	단계 3: 설계 이후
유지 보수 <i>Effort</i> (새로운 코드 추가, 삭제, 변경 등)	응용 포인트	ACT(연평균 변경 비율 (Annual Change Traffic), SU, UNFM 등의 함수.	ACT, SU, UNFM 등의 함수
노력 예측 공식 ($E=bS^c$)에서 c 의 값	1.0	선행작업, 적응도, 초기 설계, 위험제거, 팀 결집력, SEI 프로세스 성숙도에 따라 0.91 ~ 1.23	선행작업, 적응도, 초기 설계, 위험제거, 팀 결집력, SEI 프로세스 성숙도에 따라 0.91 ~ 1.23

COCOMO II 세 가지 단계

비교대상	단계 1: 응용합성 (프로토타이핑)	단계 2: 초기 설계 (7개 정도)	단계 3: 설계 이후 (17개 정도)
프로덕트 비용승수	없음	복잡도, 재사용 요구도 (Required Reusability)	신뢰도, 데이터베이스 규모, 문서화 요구정도, 재사용 요구도, 제품 복잡도
플랫폼 비용승수	없음	플랫폼 난이도	실행시간 제약, 기억공간 제약, 가상기계
인력 비용승수	없음	개인 능력과 경험	분석 능력, 응용 경험, 프로그래머 능력, 프로그래머 경험, 언어 및 도구사용 경험, 연속성
프로젝트 비용승수	없음	개발 기간, 개발 환경에 대한 요구	소프트웨어 도구 사용, 개발 기간, 여러 사이트 개발 요구

Example: 응용합성 모델로 노력 계산

1. 화면(screen), 보고서(report), 3세대 컴포넌트 숫자를 센다. 3GL 컴포넌트 예) Middleware, GUI Handling Module, Application Logic Specific Modules (for functions)
2. 화면 및 보고서 복잡도 수준 결정 (table 3.4(a))
3. 화면, 보고서, 3세대 컴포넌트의 복잡도 수준에 따른 가중치 결정. 화면, 보고서, 3세대 컴포넌트의 개수 x 가중치의 합을 구해서, OP (or AP) 계산
4. $NOP = OP \times (100 - \text{reuse}) / 100$
5. OP 생산성 (PROD)를 구함
6. $PM \text{ (person month)} = NOP / PROD$

참조: Composition Estimation Model (COCOMO II)

<https://www.geeksforgeeks.org/software-engineering-application-composition-estimation-model-cocomo-ii-stage-1/>

표 3.4(a) ▶ 화면과 보고서의 복잡도 수준

화면				보고서			
포함된 뷰의 개수	자료 테이블의 개수			포함된 섹션 의 개수	자료 테이블의 개수		
	<4	<8	8+		<4	<8	8+
<3	단순	단순	중간	0 또는 1	단순	단순	중간
3-7	단순	중간	복잡	2 또는 3	단순	중간	복잡
>8	중간	복잡	복잡	4 이상	중간	복잡	복잡

표 3.4(b) ▶ 복잡도 가중치

객체 타입	복잡도 가중치		
	단순	중간	복잡
화면	1	2	3
보고서	2	3	8
3세대 언어 컴포넌트			10

표 3.4(c) ▶ 객체 점수 생산성

개발자의 경험, 능력 개발 도구 경험, 성능	매우 낮 음	낮음	중간	높음	매우 높 음
PROD (NOP/Month)	4	7	13	23	30

각 항목의 생산성 값을 구한 후, 이
를 평균해 사용.

Example System to be built:

An **airline sales system** is to be built in C:
– Back-end database server has already been built.

- Application will have 3 screens and will produce 1 report:
 - A booking screen: records a new sale booking
 - A pricing screen: shows the rate for each day and each flight
 - An availability screen: shows available flights
 - A sales report: shows total sale figures for the month and year, and compares figures with previous months and years

The **booking screen** requires **3 data tables**, namely, the table of customer contact details, the table that records the past history of the customer, and the table of available time slots.

Only 1 view of the screen is enough. So, the booking screen is classified as **simple**.

Similarly, the levels of difficulty of the **pricing screen**, the **availability screen** and the **sales report** are classified as **simple**, **simple**, and **medium**, respectively. There is **no 3GL component**.

Table 1 Ratings for IIST airline sales system

Name	Objects	Complexity	Weight
Booking	Screen	Simple	1
Pricing	Screen	Simple	1
Availability	Screen	Medium	2
Sales	Report	Medium	3
		Total	7

The assessment on the developers and the environment shows that the developers' experience is very low (4) and the CASE tool is low (7). So, we have a productivity rate of 5.5.

According to COCOMO II, the project requires approx. 1.27 (= 7/5.5) person-months.

Effort Estimation for Early Design and Post Arch. Model

- **NOMINAL SCHEDULE ESTIMATION EQUATIONS (NS)**
for The Early Design and Post-Architecture model

$$PM_{NS} = A \times \text{Size}^E \times \prod_{i=1}^n EM_i$$

where $A = 2.94$ (for COCOMO II.2000)

where $E = B + 0.01 \times \sum_{j=1}^5 SF_j$

where $B = 0.91$ (for COCOMO II.2000)

$$TDEV_{NS} = C \times (PM_{NS})^F$$

where $F = D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j$

where $F = D + 0.2 \times (E - B)$

EM = effort multipliers (n=7 for the Early Design model, n=17 for the Post-Architecture model).

Size ← FP (Functional Point)

● Scaling Factor (Scaling Cost Driver)

Scale Factors (W_i)		ar
PREC	If a product is similar to several previously developed project, then the precededtedness is high	}
FLEX	Conformance needs with requirements / external interface specifications, ...	
RESL	Combines Design Thoroughness and Risk Elimination	}
TEAM	accounts for the sources of project turbulence and entropy because of difficulties in synchronizing the project's stakeholders.	
PMAT	time for rating: project start. Two ways for rating: 1. by the results of an organized evaluation based on the SEI CMM, 2. 18 Key Process Areas in the SEI CMM.	}

TEAM: 이해 당사자들(개발자, 고객, 사용자, 유지보수자 등)간의 응집성 (coherence)
-목적일치여부, 협업경험여부 등

PMAT: 프로세스 성숙도

기능 점수 방법

- 기능 점수(function points)

- 정확한 라인수는 예측 불가능

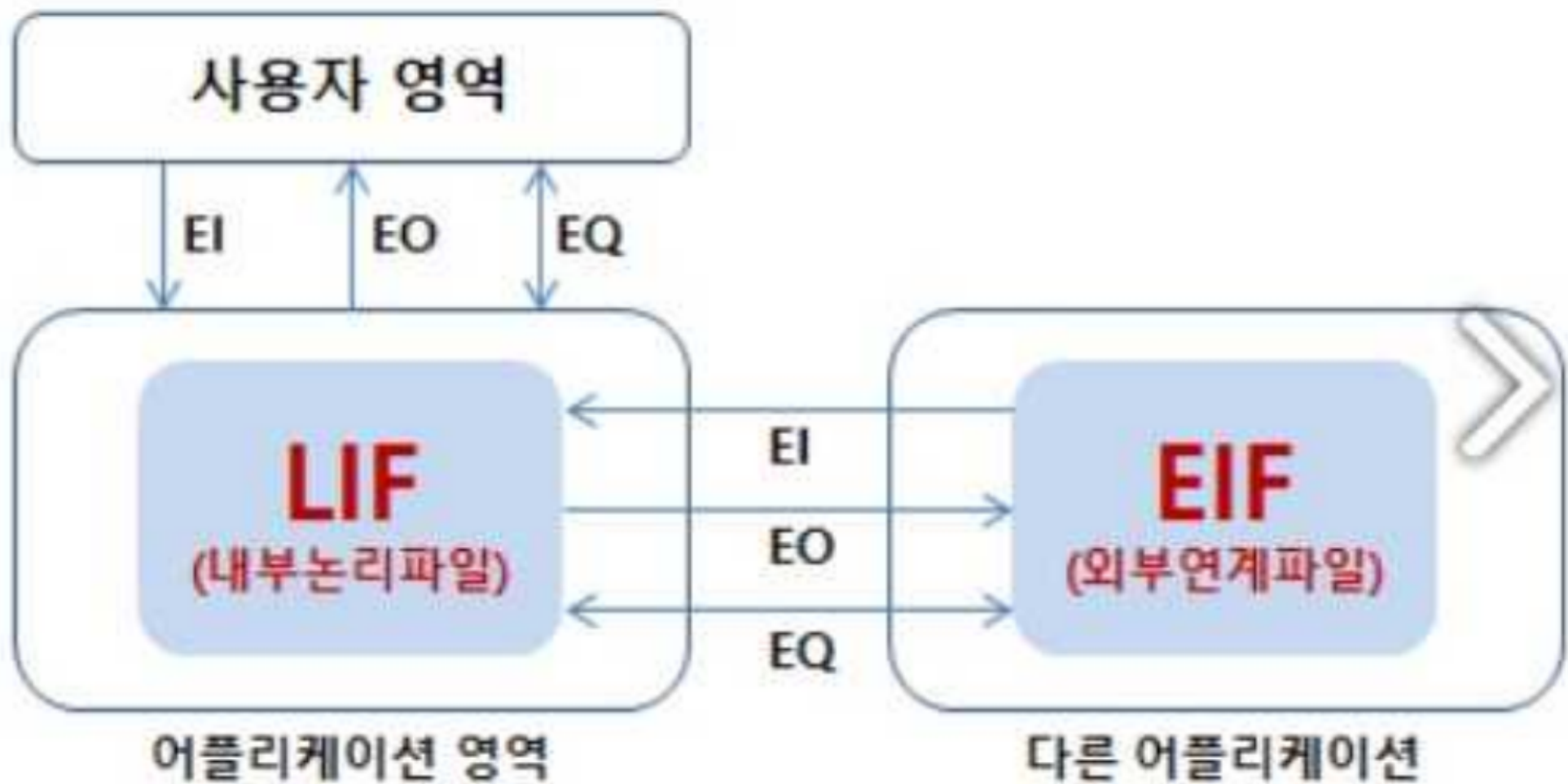
- 트랜잭션 기능: 입력(EI), 출력(EO:화면, 리포트), 질의(EQ)

데이터 기능: 파일(ILF: 내부논리파일), 인터페이스(EIF: 외부 인터페이스 파일)

- 데이터 /트랜잭션 기능 개수로 소프트웨어의 규모를 나타냄.

- 입력은 추가/수정/삭제 고려, 사용자 직접 입력 또는 파일/통신을 통한 입력 고려.

- EIF는 타시스템이 만든 파일 또는 통신라인을 통한 파일



기능점수 구성도, 이미지 출처 : SW 대가산정 가이드(2022년도 개정판)

유형	기능	내용
데이터 기능	내부논리파일(ILF)	경계 내에서 유지되는 데이터 및 제어정보 / 외부입력에 의해 유지, 모든 내부논리파일은 적어도 한 개의 외부입력을 가짐 → 예) 고객 구매이력DB, 매출DB
데이터 기능	외부연계파일(EIF)	어플리케이션 경계 밖에서 유지 데이터로 측정되는 어플리케이션 외부에서 참조하는 데이터 그룹 → 예) 고객DB
트랜잭션 기능	외부입력(EI)	어플리케이션 안으로 들어오는 데이터나 제어정보를 처리하는 단위 프로세스 / 입력, 수정, 삭제 각각 계산
트랜잭션 기능	외부출력(EO)	외부 출력(보고서 출력), 어플리케이션 경계 밖으로 보냄 / 계산 데이터 생성(자동 채번), 시스템 동작 변경이 일어나는 위치
트랜잭션 기능	외부조회(EQ)	외부 조회(리스트 조회) 유형, 데이터 가공 없이 입출력 / 파생 데이터 없음, 계산 데이터 생성하지 않음, 수학적 공식 측정 없음

로그인 기능은 EO? EQ?

- 로그인기능은 EO일수도, EQ일수도 있다.

- "EO인 경우"

- 로그인과 동시에 접속기록 등을 남기는 기능이 있는 경우



- "EQ인 경우"

- 단순히 로그인 기능

기능 점수 기본 개념

표 3.5 ▶ 기능 점수 구하는 표

	기능 분야	개수	복잡도			FP=개수X가 중값
			단순	보통	복잡	
1	입력(트랜잭션)		3	4	6	
2	출력(화면 및 출력 양식)		4	5	7	
3	질의		3	4	6	
4	내부 논리 파일 (ILF)		7	10	15	
5	외부 인터페이스 파 일(EIF)		5	7	10	
					GFP	

각 기능별로 단순, 보통 복잡을 판단하는 기준은, 표 3.7(a) 참고

기능 점수 방법

- 복합 가중값을 이용한 기능점수 산출(표 3.5)
- 총 라인수 = $FP * \text{원하는 언어의 1점 당 LOC}$
 - 기능 점수 1을 구현하기 위한 LOC
 - 어셈블리 언어(324), C언어(150), Pascal(91), Ada(71), APL(32)
- 개발 노력 = 총라인수 / 생산성(LOC/MM)

기능 점수 기본 개념

- 기능 점수는 총 기능 점수(Gross Function Point)와 처리 복잡도 보정 계수(Processing Complexity Adjustment)를 곱한 것이다.

$$FP = GFP \times PCA$$

$$GFP = \sum_{i=1}^5 (Count_i \times Complexity_i)$$

- 기능 점수는 구현되는 언어에 관계없는 메트릭.
- 기능 점수 만 사용하면 모든 항목에 일률적인 가중치가 적용되므로 문제가 있을 수 있음. ← 단순처리가 따르는 입력 와 복잡한 처리가 따르는 입력 구분 없음

기능 점수 구하는 방법

1. 다섯 가지 기능 분야에 해당되는 개수를 파악
2. 다섯 각 기능에 대한 복잡도(단순, 중간, 복잡)를 결정
3. 각 기능 분야의 개수와 복잡도 가중치를 곱하여 총 기능 점수(GFP)를 구한다.

$$GFP = \sum_{i=1}^5 (Count_i \times Complexity_i)$$

4. 14개의 질문을 이용하여 각 처리 복잡도의 정도에 따라 0에서 5까지 할당한다. (표 3.6참조)
5. 처리 복잡도 보정계수(PCA)를 다음 식을 이용하여 구한다.

$$PCA = 0.65 + 0.01 \sum_{i=1}^{14} PC$$

1. 다음 식에 넣어 기능 점수를 구한다. $FP = GFP \times PCA$

표 3.6 PCA(처리 복잡도 보정계수)

특성	처리 복잡도
① 시스템이 신뢰도 높은 백업과 복구를 요구하는가?	
② 데이터 통신이 필요한가?	
③ 분산 처리 기능이 필요한가?	
④ 성능(응답속도, 처리율 등)이 중요한가?	
⑤ 시스템이 과부하 운용 환경에서 실행되는가?	* 처리 복잡도
⑥ 온라인 데이터 입력이 필요한가?	- 0: 영향없음
⑦ 온라인 입력이 다중 화면 을 사용하는 입력 트랜잭션인 경우는 얼마나 되는가 ?	- 1: 미미 - 2: 약간
⑧ 마스터 파일이 온라인으로 갱신되어야 하나?	- 3: 보통
⑨ 입력, 출력, 파일, 질의가 복잡한가?	- 4: 상당
⑩ 내부 처리가 복잡한가(제어구조,예외처리,수학식 등 측면) ?	- 5: 많음
⑪ 재사용 되도록 설계 하여야 하나?	
⑫ 변환과 설치가 용이하도록 설계에 포함되어 있나?	
⑬ 다중 사이트에 설치되기 위하여 설계되었나?	
⑭ 변경과 사용이 쉽도록 설계되었나?	

정보통신연구진흥원 (nipa) 기능점수 산정 가이드

- 정보통신연구진흥원 (nipa) 기능점수 산정 가이드

- 기능 유형별 복잡도 판별표 (표 3.7a 참조) → 단순, 중간, 복잡

1차 기준: 레코드 타입개수(ILF, EIF) 또는
파일 타입 개수(EQ, EI).

레코드 타입 (RET라고 부름) → Table 을 말함

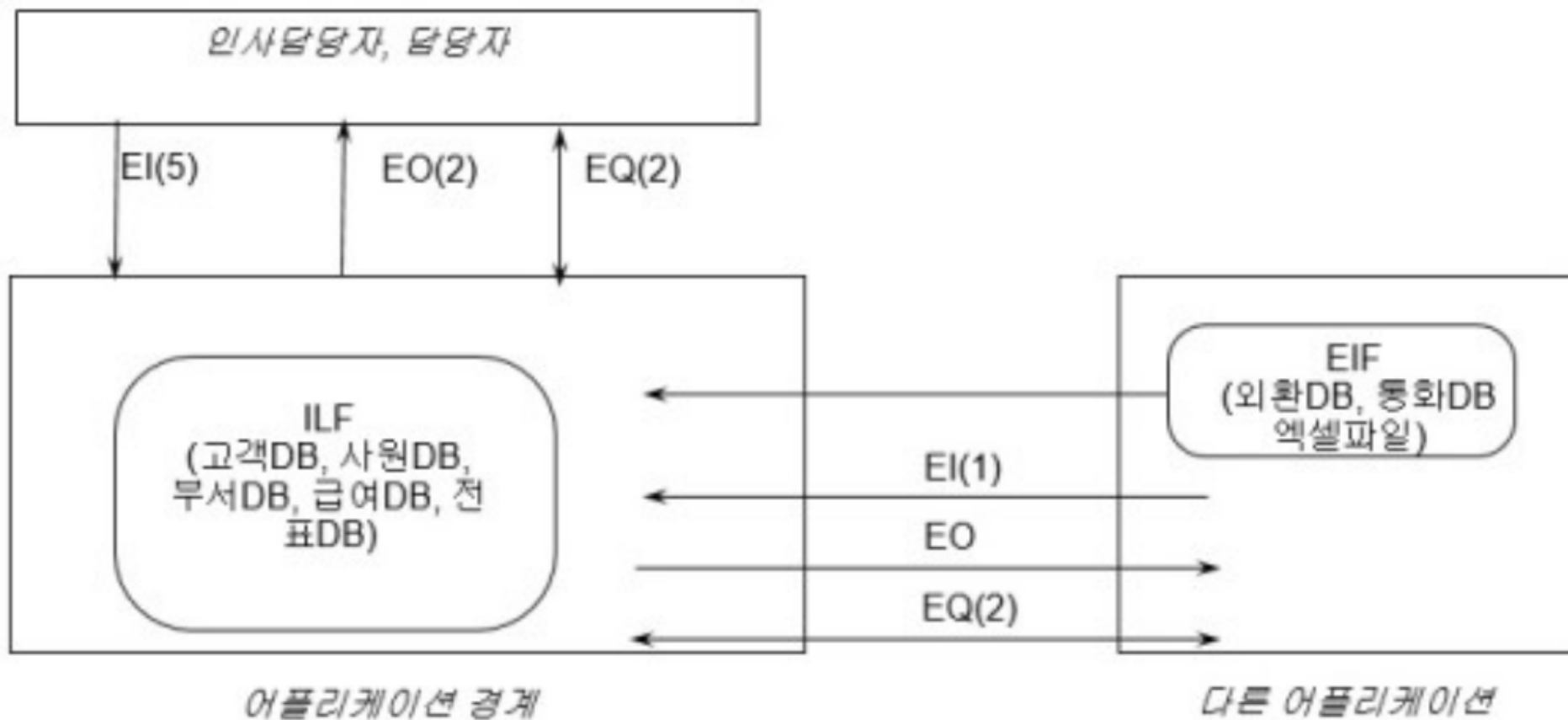
파일 타입 (FTR 라고 부름) → ILF와 EIF를 말함

2차 기준 :자료요소(DET 라고 부름)의 개수

자료요소 → 단위 데이터 항목. Table의 각 field

- 기능 유형별 복잡도에 대한 가중치 (표 3.7b 참조)

기능 점수를 이용한 노력 추정 예제



- 계산 유형은 개발프로젝트의 규모산정으로 가정한다.
- 계산범위는 내부시스템으로 회계, 인사, 고객관리, 급여, 전표시스템으로 한다.
외부시스템으로는 외환, 통화관리시스템, 외부화일은 엑셀 파일을 사용한다

기능	EI	EO	EQ	ILF	EIF
가. 담당자는 고객주문을 입력, 수정, 삭제	3			고객DB	
나. 인사담당자는 사원목록을 부서단위로 조회			1	사원DB , 부서DB	
다. 인사담당자는 사원목록을 단순 출력			1	사원DB	
라. 인사담당자는 일정 금액 이상의 급여 수령자 사원 목록을 검색		1		사원DB , 급여DB	
마. 원화에 대한 미국 달러(USD) 가치를 찾기 위해 A 은행 외환DB에서 환율을 검색			1		외환DB

기능	EI	EO	EQ	ILF	EIF
바. 인사담당자는 5년 경력 이상이고, 해당 직무 수행경험이 있는 사원목록을 추출		1		사원DB	
사. 인사담당자는 신입/경력사원 입사 시 사원 파일을 갱신	1			사원DB	
아. 인사담당자는 외국 사원 입사 시 사원의 급여를 결정하기 위해 H연합회 통화정보를 참조			1	사원DB, 급여DB	통화DB
자. 전표번호(부서번호 중 앞자리 2+년도+일련번호)를 자동 채번	1			전표DB	
차. 인사담당자는 사원 현황을 엑셀 파일로 업로드	1			사원DB	엑셀파일
계	6	2	4	5	3

GFP 계산

간이 계산법에서는, 복잡도를 낮음-보통-높음으로 구분하지 않고, 한 개 값을 사용함.

구 분		갯수	복잡도	기능점수 계산
데이터 기능점수	ILF	5	7.4	$5 \times 7.5 = 37.5$
	EIF	3	5.5	$3 \times 5.5 = 16.5$
트랜잭션 기능점수	EI	6	4.3	$6 \times 4.3 = 25.8$
	EO	2	5.4	$2 \times 5.4 = 10.8$
	EQ	4	3.8	$4 \times 3.8 = 15.2$
미조정 기능점수				데이터기능점수 + 트랜잭션 기능점수 = 105.8

기능 점수와 생산성을 이용한 노력 추정 예제

● 파악된 기능

- 사용자 입력 = 10개, 사용자 출력 = 5개, 사용자 질의 = 8개, 자료 파일 = 30개, 외부 인터페이스 = 4개, 복잡도는 모두 단순

● 처리 복잡도

- 신뢰도 매우 높은 백업, 사용 친근성은 매우 높기 요구, 나머지는 보통

● 생산성: 60 FP/week

● 해답

- [표 3.5]에 대입하여 GFP를 구한다.

$$GFP = 10 \times 3 + 5 \times 4 + 8 \times 3 + 30 \times 7 + 4 \times 5 = 304 \text{ FP}$$

- 처리 복잡도 보정 계수 구하면

$$PCA = 0.65 + 0.01(12 \times 3 + 2 \times 5) = 1.11$$

- FP를 보정

$$FP = GFP \times PCA = 304 \times 1.11 = 337.44 \text{ FP}$$

- 추정 노력(E) = $FP / \text{생산성} = 337.44 / 60 = 5.624 \text{ persons-week}$

Example of FPA

■ An inventory system that needs to

- 'Add a record'
- 'Delete a record',
- 'Display a record',
- 'Edit a record', and
- 'Print a record'
- will have
 - 3 external input types
 - 1 external output type
 - 1 external inquiry type

국내 기능 점수 산정 가이드

- 정보통신연구진흥원(NIPA)의 소프트웨어 공학에서 산정 기준을 제시[소프트웨어공학 센터, 2010]
- 산정 기준의 큰 틀은 COCOMOII의 초기 설계 모델을 따른다.
 - 외부 입력(External Input)
 - 외부 출력(External Output)
 - 내부 논리 파일(Internal Logical File)
 - 외부 인터페이스 파일(External Interface File)
 - 외부 조회(External Query)

