

## IT 프로젝트에서 개발방법이 기능점수에 미치는 영향 - 신규개발과 보강개발을 중심으로 -

이 원 호\*

### Development Method's Impact on Function Points in IT Projects

- Focused on New Development and Enhancement -

Wonho Lee\*

#### 요 약

기업이 새로운 시스템을 개발하거나 기존의 시스템을 강화해야하는 경우에 IT 프로젝트의 의사결정자들은 자체 개발과 솔루션구매 방법 사이에서 가장 최선의 개발방법을 선택해야하는 상황에 직면하게 된다. 이러한 고민을 하고 있는 프로젝트 관련 의사결정자들에게 본 연구는 프로젝트개발에 소요되는 시간, 노력, 비용 등을 측정하는데 중요한 척도가 되는 프로젝트의 기능점수에 초점을 맞추어, 내부에서의 자체개발 방식, 외부에서 개발된 솔루션구매 방식, 그리고 이러한 두 개의 방식을 혼합한 복합방식으로 개발했을 때 각각 기능점수에 어떻게 영향을 미치는지를 ISBSG에서 제공하는 프로젝트 산출물들을 통하여 의미 있는 결과들을 제공한다.

#### Abstract

The decision makers of IT projects are faced to situations to choose a best development method between inhouse development and solution purchase when companies must develop a new system or enhance the existed system. For those decision makers related to the IT projects, this study provides them with meaningful results through the ISBSG project repository, how each development method, such as a development inhouse, a solution purchased, a combined method, impact the function point, which is a critical factor to measure the required time, efforts, cost of IT project development.

▶ Keyword : 기능점수(function points), 솔루션구매(solutions purchased), 프로젝트 개발방법(project development method)

---

• 제1저자 : 이원호

• 투고일 : 2009. 02. 09, 심사일 : 2009. 03. 11, 게재확정일 : 2009. 04. 01.

\* 경북대학 인터넷정보과 부교수

## I. 서론

기존에 IT 시스템을 운영하고 있는 기업은 생산성과 경쟁력을 유지하기 위해 끊임없이 새로운 IT 시스템을 개발하거나 기존의 IT 시스템을 보강해야하는 현실에 직면하게 된다. 이러한 상황에서 IT 시스템 개발에 관련된 의사결정자는 프로젝트를 성공적으로 구축하고 운영하기 위해 자체개발과 솔루션구매 방법 사이에서 많은 고민을 하게 된다. 최근에는 금융권을 중심으로 많은 기업 사이에서 ERP 도입과 SI(System Integration) 개발 방식 간 선택문제가 주요 IT 프로젝트 개발 현안으로 떠오르고 있다. ERP는 대표적인 기업 솔루션으로 도입 초기에는 대기업을 중심으로 구축되기 시작하여, 현재는 중소기업 및 대학으로 확산되는 추세이다. ERP를 처음 공급하기 시작한 SAP는 기업이 ERP에 맞추어야하는 전략을 가지고 있는데 반해, 경쟁 공급업체인 Oracle은 기업의 요구에 맞추어 ERP를 커스터마이징(customizing)하는 전략으로 차별화하고 있다. 즉, SAP는 검증되고 표준화된 ERP에 맞추어 기업의 기존 업무 프로세스를 리엔지니어링(re-engineering)하려는 고객에 초점을 맞춘 반면, Oracle의 솔루션 판매 전략은 ERP와 SI 사이에서 고민하고 있는 고객들을 대상으로 기업이 원하고 요구하는 업무 프로세스에 융통성 있게 ERP를 맞추려는 전략으로 접근하고 있다. 이러한 SAP 전략은 본 연구의 대상인 솔루션 구매 방식과 유사한 반면 Oracle의 전략은 솔루션 구매에 자체개발의 장점을 혼합한 복합개발 방식과 동일한 맥락을 가진다고 보여진다. ERP이외에도 CRM, SCM, BPM, KMS, BI 등 여러 종류의 기업용 솔루션의 도입과정에서도 이와 유사한 상황에 접하게 될 것으로 보인다. 물론, 기업이 내부 업무 특성에 맞춘 SI방식으로 자체 개발하는 경우에도 솔루션 도입을 안 함으로서 발생할 수 있는 또 다른 반대의 위험에 직면할 수 있고, 이에 따라 자체개발에서 발생할 수 있는 문제점을 일부 솔루션 구매방식을 통해 보완하려는 복합방식 개발이 선택되어 질 수도 있다. 이렇게 기업이 프로젝트를 개발할 때 직면하게 되는 복잡한 상황에서 어떠한 개발 방법이 가장 우위에 있다고 판단하기에는 아직 이와 관련된 연구결과도 미비하고 이를 검증할 측정 수단도 마련되어 있지 않아 어렵다고 판단된다. 이에 본 연구는 IT 프로젝트 개발 방법의 중요한 선택 기준인 개발 소요시간, 비용, 성과 등의 간접적인 측정 수단의 근간이 될 수 있는 IT 프로젝트의 기능점수에 초점을 맞추어, 내부에서의 자체개발 방식, 외부에서 개발된 솔루션구매 방식, 그리고 이러한 두 개의 방식을 혼합한 방식이 각각 기능점수에 어떻게 영향을 미치는지를 연구하여 IT 프로젝트 개발에 관련

된 의사결정자들에게 프로젝트 유형에 따른 개발방법 선택 시 합리적인 판단을 할 수 있는 근거자료를 제공해 주려고 한다.

## II. 선행연구

### 2.1 기능점수 관련 선행연구

기능점수(FP : Function Point)는 기존의 LOC (Line of Code) 같은 개발자 중심의 물리적 접근 방식에서 벗어나, 사용자 관점에서 경제적 가치와 트랜잭션 기능, 데이터기능과 같은 사용자의 요구기능을 고려한 소프트웨어 규모 및 비용 산정 파악을 위한 표준방식이다.[1] 이러한 기능점수는 많은 선행연구[2][3][4]에서 나타난 것과 같이 IT 프로젝트의 노력 품질, 생산성, 규모, 비용(개발비용, 유지보수비용 등)산정 등을 측정하기 위한 유용한 간접지표로 많이 신뢰되며 활용되어지고 있다[5][6][7]. 본 연구에서는 이와 같은 문헌연구를 토대로 기능점수를 IT 프로젝트의 품질, 생산성을 나타내는 간접척도로 간주하여, 기능점수를 통하여 프로젝트 개발방법 간의 결과를 비교분석하도록 한다.

본 연구에서 사용할 기능점수의 근간이 되는 Albrecht 모형[8]의 다섯 가지 기능인 외부입력(EI : External Input), 외부출력(EO : External Output), 내부논리파일(ILF : Internal Logical File), 외부인터페이스파일(EIF : External Interface File), 외부조회(EQ : External Inquiry)의 개념을 정리하면 <표 1> 과 같다.

표 1. Albrecht 모형의 기능 및 개념  
Table 1. function and concept of Albrecht model

기능	개념
외부입력	시스템의 경계 밖에서 시스템 데이터를 입력하는 기능
외부출력	시스템의 경계 밖으로 데이터를 출력하는 기능
내부논리파일	시스템 내에서 논리적으로 유지되는 파일로서, 사용자 관점에서 볼 때 시스템에 의해 생성되고 사용되며 관리되는 논리적인 데이터 파일
외부인터페이스파일	응용시스템간에 공유되거나 전송되는 파일로서 시스템 밖의 응용시스템이 논리적으로 유지하는 데이터 파일
외부조회	입력이 출력을 즉시 요구하는 입/출력 조합

기능 유형별 복잡도를 단순(Low), 보통(Average), 복잡(High)의 3단계로 나누어 가중치 계산을 하는 기능점수(UFP : Unadjusted Function Point) 산출 방식은 <표 2>와 같다.[9]

표 2. 복잡도를 가중한 기능점수(UFP) 계산  
Table 2. UFP Calculation weighted with complexity degree

기능유형	복잡도	가중치	기능유형별 합계
내부논리파일	단순(Low)	_____ × 7	
	보통(Average)	_____ × 10	
	복잡(High)	_____ × 15	_____
외부인터페이스 파일	단순(Low)	_____ × 5	
	보통(Average)	_____ × 7	
	복잡(High)	_____ × 10	_____
외부입력	단순(Low)	_____ × 3	
	보통(Average)	_____ × 4	
	복잡(High)	_____ × 6	_____
외부출력	단순(Low)	_____ × 4	
	보통(Average)	_____ × 5	
	복잡(High)	_____ × 7	_____
외부조회	단순(Low)	_____ × 3	
	보통(Average)	_____ × 4	
	복잡(High)	_____ × 6	_____
총 기능점수 합계 (조정 전)			_____

〈표 2〉의 UFP 산출과정에서 적용되는 복잡도는 동일한 복잡도라도 기능유형에 따라 그 가중치가 다르며, 복잡도의 판단은 〈표 3〉과 같은 매트릭스(matrix)를 기준으로 한다. 〈표 3〉에서 표시된 DET(Data Element Type)는 사용자가 인지할 수 있고 반복되지 않는 유일한 필드의 수이고, RET(Record Element Type)는 사용자가 인지하는 데이터요소들의 서브그룹(subgroup) 수이다.

표 3. 복잡도 매트릭스  
Table 3. Complexity Matrix

RET \ DET	1~19	20~50	51 이상
1	Low	Low	Average
2~5	Low	Average	High
6 이상	Average	High	High

〈표 4〉에 나타난 것과 같이 시스템의 일반적인 14개의 항목에 대한 영향도(DI : Degree of Influence)를 0에서 5까지 평가 척도를 가지고 결정한 후, 각각의 산출된 DI 값을 합산한 총 영향도(TDI : Total Degree of Influence)를 가지고 아래 절차와 같이 값조정인자(VAF : Value Adjustment Factor)를 계산한다.[9]

- ① step 1 : DI(0 ~ 5)를 결정하기 위해 14개의 GSC (General System Characteristics)를 측정한다.
- ② step 2 : TDI를 산출하기 위해 14개 모든 GSC의 DI를 더한다.
- ③ step 3 : 계산된 TDI 값을 아래 공식에 대입하여 VAF를 산출한다.

$$VAF = (TDI * 0.01) + 0.65$$

일반적으로 이러한 절차를 적용했을 경우 VAF는 UFP에서 플러스(+) 또는 마이너스(-)로 35 % 정도 범위 내에서 조정된다.

표 4. 총 영향도(TDI)산출을 위한 14개의 시스템 항목  
Table 4. 14 system items for output of total TDI

시스템 항목 (특징)	영향도(DI : Degree of Influence) 0-5
1. 데이터 통신	_____
2. 분산데이터 처리	_____
3. 시스템 성능	_____
4. 사용여유 정도	_____
5. 트래잭션 비율	_____
6. 온라인 데이터 입력	_____
7. 최종 사용자 효율성	_____
8. 온라인 갱신정도	_____
9. 처리 복잡성	_____
10. 재사용성	_____
11. 설치 용이성	_____
12. 운영 용이성	_____
13. 다중 설치성	_____
14. 유지보수 용이성	_____
총 영향도 (TDI) : Total Degree of Influence)	_____

의 기능점수(FP)는 <표 2>에서 산출된 조정 전 기능점수(UFP)와 <표 4>에서 산출된 총 영향도(TDI)로 계산되어진 값 조정인자(VAF)에 의해 다음 수식과 같이 최종 산출되게 된다.[9]

$$FP = UFP * VAF$$

본 연구에서는 기능유형별 복잡도를 가중한 기능점수(UFP)에 총 영향도(TDI)로 계산되어진 값조정인자(VAF)를 반영한 최종 기능점수(FP)를 대상으로 한다.

<표 5>는 본 연구의 근간이 되는 Albrecht 기능 모형과 다른 연구된 모형들의 프로젝트의 세부 요소를 비교한 것이다.

표 5. 기능점수 관련 IT 프로젝트 요소  
Table 5. IT project factors related to FP

Albrecht(8) 기능점수 모형	<ul style="list-style-type: none"> <li>• 기 능 요 소 : 사용자 입력 수, 사용자 입출력 수, 파일 수, 외부 인터페이스 수</li> <li>• 복잡도 요소 : 데이터 통신, 분산처리, 시스템 성능, 시스템 여유, 처리량, 온라인 데이터 입력, 트랜잭션, 온라인 갱신, 처리 복잡성, 재사용, 설치 용이성, 운용상 편리함, 다수의 장소의 설치성, 변경 용이성</li> </ul>
Boehm(10) COCOMO 모형	<ul style="list-style-type: none"> <li>• 프로젝트 속성 : 최신프로그래밍 사용기술, 소프트웨어 도구 사용, 요구되는 개발기간</li> <li>• 프로그래밍 속성 : 요구되는 소프트웨어 신뢰도, DB 규모, 프로젝트 복잡도</li> <li>• 요원 속성 : 분석가 능력, 응용시스템 경험, 프로그래머 능력, 가상기계 경험</li> <li>• 컴퓨터 속성 : 실행시간 제약, 주지역장치 제약, 가상기계 휘발성</li> </ul>
Jones(11) 특성점수	<ul style="list-style-type: none"> <li>• 특성요소 : 사용자 입력 수, 사용자 출력 수, 사용자 입출력수, 파일수, 외부 인터페이스 수, 알고리즘 수</li> </ul>

### 2.2 개발방법 관련 선행연구

기존의 IT 프로젝트는 현업 사용자의 요구사항에 따라 내부에서 보유하고 있는 COBOL, C, Java, Visual Basic, Power builder 등의 개발언어로 자체개발을 해왔으나, 최근에는 ERP, SCM, CRM, BI 등의 새로운 기업 솔루션이 등장하면서 프로젝트 개발에 도입하는 사례가 많이 늘어났다.[12]

자체개발(development inhouse)의 장점으로는 시스템의

사용목적은 사용자 자신이 가장 잘 알기 때문에 자체인력으로 자신이 사용할 시스템을 구현가능하다는 점[13]과 업무변경 또는 사용자의 수요요구 사항을 탄력적으로 융통성 있게 수용할 수 있다는 점이다. 반면, 자체개발은 소프트웨어의 낮은 기능 수준, 긴 개발시간, 표준화 및 안정성 등에서 문제가 될 수 있다.[14] 이와는 반대로, 솔루션구매(solution purchased)는 자체개발이 가지고 있던 문제점들을 극복할 수 있고 [15], 솔루션이 제시하는 전문성과 성공적 사례의 업무 템플릿을 활용[14]할 수 있는 반면, 기업의 업무 프로세스에 맞도록 솔루션 패키지(package)의 기능을 수정하는데 소요되는 불확실한 시간과 비용[15], 솔루션 공급업체에 관련된 불확실성, 사용자 요구에 대한 패키지 기능의 대응력 등을 정확하게 파악하기 힘들다는 점 등의 단점을 지니고 있다. 이와 같이 자체개발과 솔루션 구매의 방식이 각각 장단점을 가지고 있다는 연구 결과를 토대로 실제로 수행된 프로젝트 결과에서는 각각의 개발방법이 어떻게 프로젝트의 성과[16]에 영향을 미쳤는지를 프로젝트의 기능점수 결과를 통해 간접적으로 검증하도록 한다. 이와 함께 프로젝트의 성과 또는 위험요인은 프로젝트 영역이 신규개발 또는 기존개발, 운영유지보수에 따라 차이가 있을 수 있다는 문헌연구 결과[17]에 착안하여, 본 연구에서도 개발유형에 따라 개발방법이 기능점수에 미치는 영향에 차이가 있을 것이라는 가설도 동시에 설정하도록 한다.

## III. 가설설정 및 연구모형

### 3.1 가설 설정

앞서 제 2장에서 논의된 내용을 기초로 설정한 가설들을 정리하면 다음과 같다.

가설 1: 개발방법에 따라 프로젝트 기능점수에 미치는 영향 정도의 차이가 있을 것이다.

가설 2 : 개발유형은 가설1에 영향을 미칠 것이다.

### 3.2 연구모형

앞서 설정한 가설에서와 같이 IT 프로젝트의 기능점수는 프로젝트 개발방법의 선택에 따라 영향을 받으며, 이러한 개발방법은 프로젝트의 개발유형에 따라서 그 결과가 다르게 나타날 것으로 예측된다. 가설 1은 IT 프로젝트 개발을 시작하기 전에 자체적인 내부개발, 외부에서 솔루션구입, 아니면 이 두 가지 개발방법을 혼합한 방식 중 어떠한 개발방법을 선택

하느냐에 따라 기능점수에 어느 정도 차이가 미치는 지를 비교 분석하여, 향후 IT 프로젝트 개발 계획을 가지고 있는 기업 또는 기관의 의사결정자에게 참고가 될 수 있는 결과를 제공하기 위해 설정되었다. 가설 2는 IT 프로젝트의 개발유형이 신규개발인지, 또는 보강개발인지에 따라 가설1의 결과가 어떻게 다르게 나타나는 지를 관찰하기 위해 설정되었다.

지금까지 설정한 가설을 기초로 관찰 대상인 개발방법 및 개발유형과 프로젝트의 기능점수와의 관계를 연구모형으로 만들어 나타내면 <그림 1>과 같다.

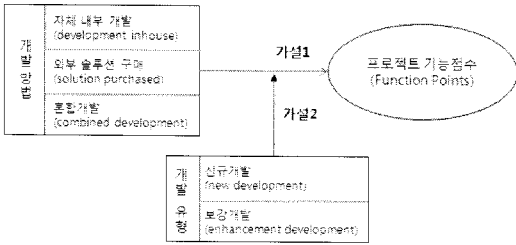


그림 1. 연구모형  
Fig 1. study model

#### IV. 연구설계

##### 4.1 연구 변수의 정의

- **기능점수** : 값 조정인자(Value Adjustment Factor)에 의해 조정된 기능점수(Function Point Count)로 IFPUG 방식에 의해 산출된 값만 추출하여 연구대상에 포함시킨다.
- **개발방법** : 프로젝트 개발방법은 솔루션구매(solution purchased)방식과, 자체개발(developed in house) 그리고 솔루션 구매와 자체 개발 방법을 혼합한 복합개발(combined)의 세 가지 종류 값으로 나타내어진다.
- **개발유형** : 프로젝트 개발형태는 기존의 시스템을 강화시키는 보강개발(enhancement development)과 새로운 시스템을 개발하는 신규개발(new development)의 두 가지 종류 값으로 나타내어진다.

##### 4.2 모집단 자료의 특징

본 연구 표본자료의 원천이 되는 모집단 자료는 ISBSG (International Software Benchmarking Standards Group)에서 제공하고 있는 축적된 자료(data repository)

로서 호주, 미국, 영국, 일본 등 세계 20 여 국가의 총 716개 프로젝트에서 수집되어 축적되어온 자료들이다. 모집단 전체 업무별 분포는 <표 6>과 같다.

표 6. 업무별 분포  
Table 6. business ratio

사업 영역	비율(%)
회계업무	5.8
은행업무	23.4
엔지니어링업무	5.6
재무업무	15.6
보험업무	7.1
재고관리	3.2
범용업무	3.4
유통업무	1.2
제조업무	7.3
마케팅	0.2
인사관리	2.9
연구개발	2.0
영업	2.5
통신업무	2.4
기타	17.5
총 계	100.0

##### 4.3 표본자료 추출방법

연구 대상이 되는 표본자료를 추출하기 위해 모집단 자료에서 IFPUG 기능점수, 개발유형, 개발방법 항목만을 필터링 하여 총 225건의 표본자료를 생성하였다. 이 과정에서 모집단 정규분포에서 벗어나는 값들(outliers) 또는 관련이 없고(irrelative), 적용할 수 없거나(not applicable) 알려지지 않은(unknown) 값들은 표본자료에서 제외시켰다.

표 7. 표본자료의 개발유형 및 방법 분포  
Table 7. development type and method ratio of sample data

개발유형		개발방법		
신규개발	보강개발	자체개발	솔루션구매	복합개발
67 %	33 %	50 %	22 %	28 %

##### 4.4 표본자료 분석방법

본 연구에서는 SPSS 12.0 통계패키지를 활용하여 두 가지 형태의 개발유형에 따라 가설1이 영향을 받는지를 우선 검증하기 위해 이원분산분석(two-way ANOVA)을 실행하였고, 다음 단계로 가설2가 유효하다는 검증결과를 토대로 표

본자료를 신규개발과 보강개발로 분리한 후에 각각의 그룹에 대해 일원분산분석(one-way ANOVA)을 실행하여 가설1을 검증하였다.

## V. 가설검증 및 결과분석

### 5.1 개발유형의 영향 여부 검증

아래 <표 8>은 가설검증을 위해 사용되는 신규개발과 보강개발이 혼합된 표본집단이 모집단과 동일한 분산을 가지고 있다는 가정을 충족시키는 지를 알아보기 위해서 실행한 Levene 검정 결과이다.

표 8. 개발유형이 혼합된 표본의 Levene 검정  
Table 8. Levene examination of mixed sample data in development type

오차 분산의 동일성에 대한 Levene의 검정\*

종속변수: Function Points

F	자유도1	자유도2	유의확률
1.325	5	208	.255

여러 집단에서 종속변수의 오차 분산이 동일한 영가설을 검정합니다.

a. 계획:  
Intercept+DevelopmentType+DevelopmentMethod

통계 결과 Levene의 F값이 1.325으로서 충분히 크며, 유의확률도 0.255로 유의수준인 0.05 보다 크므로 모집단의 분산이 동일하다는 귀무가설이 채택된다. 이후, 이원분산분석(two-way ANOVA)을 실행하여 다음 <표 9>와 같이 나타난 결과, 개발유형(development type)의 F값은 6.671 이고, 유의확률은 0.010 으로 유의한 것으로 나타났다. 이와 같은 결과로, 개발유형에 따라 개발방법이 기능점수에 미치는 영향 정도의 차이가 있다는 가설2가 채택되어지고, 이를 근거로 표본자료를 개발유형인 신규개발과 보강개발로 각각 분리하여 실행한 후 가설1을 검증한다.

표 9. 개발유형의 영향 검정  
Table 9. impact examination of development type

개체-간 효과 검정

종속변수: Function Points

소스	제1 유형 제곱합	자유도	평균제곱	F	유의확률
수정 모델	2703173.4*	3	901057.813	2.947	.034
절편	55027754	1	55027754	180.001	.000
DevelopmentType	2039298.9	1	2039298.9	6.671	.010
DevelopmentMethod	663874.551	2	331937.275	1.086	.340
오차	64198796	210	305708.552		
합계	121929723	214			
수정 합계	66901969	213			

a. R 제곱 = .040 (수정된 R 제곱 = .027)

### 5.2 개발유형별 가설검증

#### 5.2.1 신규개발에서의 통계결과 분석

표본집단을 생성하는 과정에서 개발방법이 자체개발(development inhouse)은 1, 솔루션구매(solution purchased)는 2, 복합개발(combined)은 3으로 각각 디코딩한 다음, 일원분산분석(one-way ANOVA)을 실행하여 <표 10>과 같은 결과를 얻었다.

표 10. 신규개발에서 FP에 미치는 차이 비교  
Table 10. Comparison of impact on FP in new development

다중 비교

LSD	종속변수: 기능점수		평균차 (I-J)	표준오차	유의확률	95% 신뢰구간	
	(I) DevMethod	(J) DevMethod				하위구간	상위구간
1	2	3	-187.312*	76.589	.001	-312.13	-162.89
	2	3	36.678	71.530	.610	-104.84	178.00
2	1	3	167.312*	76.589	.001	15.89	318.73
	2	3	203.890*	77.863	.010	49.95	357.83
3	1	2	-36.578	71.530	.610	-178.00	104.84
	1	3	-203.890*	77.863	.010	-357.83	-49.95

\*. 1. 수준에서 평균차가 분다.

<표 10>의 LSD 검정결과에 의하면 유의수준 0.05에서 자체개발과 솔루션구매의 기능점수 평균차이가 유의하게 나타났다. 특히 솔루션구매와 복합개발의 기능점수의 평균차이는 유의수준 0.05하에서 두드러짐을 알 수 있다. 그러나 자체개발과 복합개발과의 기능점수의 평균차이는 통계결과 유의하지 않는 것으로 나타났다. 아래 <그림 2>는 신규개발에서 각 개발방법 기능점수의 평균을 도표로 나타낸 것으로, 솔루션구매가 다른 개발방법보다 현저하게 높게 표시되고 있다.

평균 도표

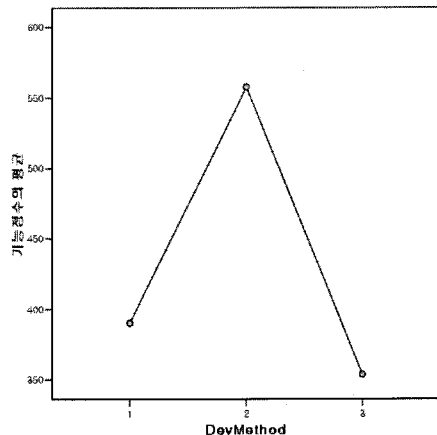


그림 2. 신규개발에서 개발방법별 FP 평균도표  
Fig 2. FP average chart for development method in new development

### 5.2.2 보강개발에서의 통계결과 분석

표본집단에서 개발유형이 신규개발을 제외한 보강개발만을 추출하여 동일한 방법으로 일원분산분석(one-way ANOVA)을 실행하여 <표 11>과 같은 결과를 얻었다.

표 11. 보강개발에서 FP에 미치는 차이 비교  
Table 11. Comparison of impact on FP in enhancement development

다중 비교

종속변수: FunctionPoint

LSD	(I) DevMethod	(J) DevMethod	평균차 (I-J)	표준오차	유의확률	95% 신뢰구간	
						하한	상한
	1	2	-23.549	36.099	.517	-96.17	49.07
	1	3	-83.438	36.099	.025	-156.06	-10.82
	2	3	23.549	36.099	.517	-49.07	96.17
	3	1	-59.889	45.102	.191	-150.62	30.84
	3	2	83.438	36.099	.025	10.82	156.06
		2	59.889	45.102	.191	-30.84	150.62

\*. 05 수준에서 평균차가 큼니다.

<표 11>의 LSD 검증결과에 의하면 신규개발과는 달리, 보강개발에서는 유의수준 0.05에서 자체개발과 복합개발에서만 기능점수 평균차이가 유의하게 나타났고, 그 외의 개발방법 간에서는 유의수준 0.05하에서 차이가 없게 나타났음을 알 수 있다. 아래 <그림 3>은 보강개발에서 각 개발방법별 기능점수의 평균을 도표로 나타낸 것으로, 앞서 신규개발에서 솔루션구매가 다른 개발방법보다 현저하게 높게 나타난 것과는 달리, 보강개발에서는 복합개발이 가장 높게 표시되고 있다.

평균 도표

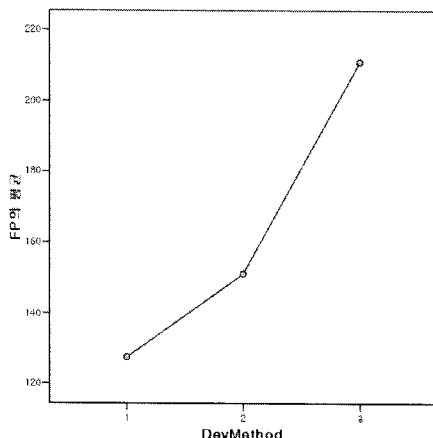


그림 3. 보강개발에서 개발방법별 FP 평균도표  
Fig 3. FP average chart for development method in enhancement development

### 5.3 결과분석

앞서 나타난 통계결과 자료를 근거로 가설 검증 결과를 요약하면 <표 12>와 같다.

표 12. 가설에 대한 검증 결과

Table 12. examination results for the hypothesis

가 설		검증 결과
가설 1	개발방법에 따라 프로젝트 기능점수의 미치는 영향 정도의 차이가 있을 것이다.	채택
가설 2	가설1은 개발유형에 따라 그 결과가 다르게 나타날 것이다. - 신규개발 : 자체개발과 솔루션구매의 기능점수 평균차이 / 솔루션구매와 복합개발의 기능점수의 평균차이가 유의하게 나타남 - 보강개발 : 자체개발과 복합개발에서만 기능점수의 평균차이가 유의하게 나타남	채택

개발유형이 신규개발 또는 보강개발이냐에 따라 가설1이 영향을 받는다는 것이 통계적으로 유의하게 나타났고, 이와 같은 결과를 토대로 프로젝트의 개발유형을 신규개발과 보강개발과 나누어 검증한 결과, <표 12>와 같이 개발유형에 따라 그 결과가 다소 다르게 나타났다. 즉, 신규개발에서는 자체개발과 솔루션구매의 기능점수 평균차이, 솔루션구매와 복합개발의 기능점수 평균차이가 각각 유의하게 나타난 반면, 보강개발에서는 자체개발과 복합개발에서만 기능점수의 평균 차이가 유의하게 나타났다. 이와 같은 결과를 근거로 신규개발에서는 솔루션구매가 자체개발과 복합개발에 비해 기능점수가 가장 높고, 보강개발에서는 복합개발 방법이 자체개발보다 기능점수가 높다고 판단할 수 있다. 이러한 연구 결과는 신규개발에서는 소프트웨어의 고기능, 짧은 개발소요시간, 표준화 및 안정성 등과 같은 장점을 가지고 있는 솔루션구매 방법이 다른 개발방법보다 우위에 있음을 나타내주고 있다. 또한 보강개발에서는 이미 구축되어 있는 시스템을 보완하는 자체개발 방법과 자체개발에서 예상되는 단점을 보완해 줄 수 있는 솔루션 구매 방식을 혼합한 복합개발 방법이 기능점수 상승에 가장 긍정적인 영향을 주고 있음을 나타내주고 있다.

## VI. 결론

IT 프로젝트 개발방법에 관련된 기존의 선행연구에서는 솔루션구매 방식이 자체개발보다는 여러 가지 장점을 가지고 있는 이유로 권장하고 있으나, 본 연구 결과에 의하면 개발유형이 신규개발에서는 이러한 연구결과와 일치되나, 보강개발에서는 오히려 솔루션구매 방식보다는 자체개발과 솔루션구매가 혼합된 형식인 복합개발이 더 효과가 있음이 입증되었

고, 개발방법 선택이 개발유형에 따라 달라져야함을 나타내 주고 있다. 또한, 본 연구결과는 기업에서 신규개발 또는 보강개발 시에 프로젝트의 기능점수를 높이기 위해 주어진 개발 유형 상황에 따라 최적의 개발 방법을 선택해야하는 결정과정에서 벤치마킹할 수 있는 통계적 근거를 제공함으로써, 프로젝트의 개발방법론 결정에 관련된 사람들에게 의미 있는 메시지를 전달해 주고 있다. 특히, 대기업에 비해 IT 프로젝트 개발경험이 전무하거나 미비한 중소기업에게는 본 연구 결과의 의미가 상대적으로 크다고 볼 수가 있다. 더 나아가서 본 연구에서 표본자료로 활용된 IT 프로젝트 자료가 국내가 아닌 세계 각국에서 제공된 것으로, 향후 해외기업을 대상으로 IT 프로젝트를 아웃소싱하거나 해외 프로젝트에 관련된 일을 수행하는 기업에게는 유용한 자료가 될 것으로 기대된다.

한편, 본 연구에서 통계자료로 활용한 프로젝트 산출 결과물들은 우리나라를 제외한 세계 20여개 국가에서 수행한 ISBSG에서 제공하는 측정데이터로 우리나라의 현실과는 다소 차이가 있을 수는 있다. 또한 프로젝트가 속하는 산업군과 업무별 특성에 따라 그 결과가 다르게 나올 가능성도 완전히 배제할 수는 없다. 그러나 IT 프로젝트 개발방법이 자체개발이든 솔루션구매 등 프로젝트 개발에 활용되는 프로그래밍언어, 패키지, 소프트웨어 등이 이미 세계 공통 개발 도구로서 사용되어지고 있고, IT 프로젝트 개발 속성장 국내 프로젝트도 해외 프로젝트와 유사한 결과가 나올 것으로 예측된다.

향후 연구과제로, 본 연구에서 대상으로 한 해외 프로젝트 결과 자료가 아닌, 순수하게 국내 프로젝트 결과 자료만을 대상으로, 해외 프로젝트와 국내 프로젝트의 통계 결과를 산출한 후 비교 분석하는 후속 연구의 필요성이 제기된다. 이와 함께 호주의 ISBSG 처럼 우리나라에도 국내에서 그 동안 수행된 IT 프로젝트의 벤치마킹 자료를 축적하고 제공해주는 기관을 설립하고, ISBSG와의 활발한 교류를 통해 IT 프로젝트와 관련된 다양한 국내외 자료제공 서비스가 이루어지기를 기대한다.

## 참고문헌

- [1] KOSMA <http://www.kfpug.or.kr>
- [2] 김현수, "기능점수를 이용한 소프트웨어 규모 및 비용산정 방법에 관한 연구," 경영과학 제14권 제1호, 131-149쪽, 1997년 5월
- [3] 김현수, "프로젝트 측정 및 관리시스템 개발연구," 쌍용정보통신, 20-25쪽, 1998년 8월
- [4] 박찬규 · 신수정 · 이현옥, "국내 소프트웨어 개발사업에 적합한 기능점수규모 예측방법에 관한 연구," 경영과학, 제20권 제2호, 179-196쪽, 2003년 11월
- [5] Albrecht, A.J. And Gaffney, J.E., "Software Function, Source Line of Code, and Development Effort Prediction: A Software Science Validation," IEEE TSE, Vol. 9, No. 6, pp. 639-648, Nov 1983
- [6] Kemerer, C. F., "Reliability of function points measurement: A field experiment," Communications of ACM, 36(2), pp. 85-97, 1993
- [7] Low, G.C., and Jeffery, D.R., "Function Points in the estimation and evaluation of the software process," IEEE TSE, 16(1), pp. 81-84, 1990
- [8] Albrecht, A.J., "Measuring application development productivity. In GUIDE/SHARE: Proceedings of the IBM Applications Development Symposium," Monterey, CA., pp. 83-92, 1979
- [9] IFPUG, "Function Point Counting Practices Manual Release 4.1," IFPUG, A2-A3, Jan. 1999
- [10] Boehm, B. W., "Soft Engineering Economics," Prentice-Hall, 1981
- [11] Jones, C., "A Short History of Function Points and Feature Points", Software Productivity Research Inc., Burlington, MA, 1986
- [12] 박송미, 채명신, "프로젝트 위험요인 인식에 관한 비교 연구," 정보시스템연구, 제16권 제4호, 243-268쪽, 2007년 12월
- [13] 강석호, "E-biz IT SI," 박영사, 2003. 3
- [14] 신현식, "정보시스템 외부조달 방법과 자체 개발 방법의 비교분석," KAIST 박사학위논문, 1995.
- [15] Lynch, R. K., "Implementing Packaged Application Software: Hidden Costs and New Challenges," Systems, Objectives, Solutions, Vol. 4. 1984.
- [16] Sawyer, S. and Guinan P. J., "Software Development: Processes and Performance," IBM Systems Journal, Vol. 37, No. 4, 1998
- [17] 오수진, "SI프로젝트 범위증가에 영향을 미치는 위험요인 인지에 대한 연구," 한양대 석사학위 논문, 2002.



## 저 자 소개



### 이 원 호

- 1988. 서강대학교 전산학 학사
- 1995. 미국 Colorado 대학교 정보시스템 석사
- 2004. 국민대학교 MIS 박사
- 1988~1996 대우증권 정보시스템부
- 1996~현재 경북대학 인터넷정보과 부교수
- 관심분야 : e비즈니스, 데이터베이스