

Perl/CGI와 PHP의 비교를 통한 웹 어플리케이션 개발성과에 미치는 영향에 관한 연구

A Study on the Effects of Server-Side Languages(Perl/CGI, PHP) on the Web Application Development Productivity and Developers' Satisfaction

정인근¹⁾, 이명무²⁾, 김용진³⁾

서울시 동대문구 이문동 270 한국외국어대학교 경영학과

초 록

본 연구의 목적은 프로그램 언어(Perl/CGI, PHP)가 웹 어플리케이션의 개발성과에 미치는 영향을 조사하기 위한 것이다. 이를 위해 본 연구는 정보시스템의 복잡성과 프로그램 언어에 따른 프로그래머의 생산성 효과 이론 등과 같은 선행연구(Wrigley & Dexter, 1991; McFarlan, 1981; He & McLean, 1985)와 연구 대상인 Perl/CGI와 PHP에 관한 이론을 조사하였다. 이를 바탕으로 프로그램 언어(Perl/CGI, PHP)가 과업의 위험성(개발자 경험, 요구사항의 난이도)에 따라 개발성과 생산성, 개발자 만족도에 미치는 영향을 조사하기 위한 모델을 개발하고 실험을 통해 이를 검증하였으며, 결과는 다음과 같다.

사용 언어에 따른 집단별(초보자집단/전문가집단) 성과의 차이는 요구사항이 간단할 경우에는 초보자집단과 전문가집단 사이에 개발성과의 차이점을 발견할 수 없었다. 그러나 요구사항이 복잡할 경우에는 두 개의 프로그램 언어 모두에 대해 전문가 집단이 초보자 집단보다 개발성과가 높게 나타났다. 또한 사용 언어별 성과에서는 PHP를 사용하는 경우가 Perl/CGI를 사용하는 경우보다 개발자 경험이나 요구사항의 난이도와 관계없이 높은 개발성과를 보여 주었다.

1. 서 론

오늘날의 정보기술은 기업, 산업 차원에서 뿐만 아니라 국가 경쟁력 차원에서 전략 변수로서 그 중요성이 더욱 커지고 있다. 그러나 이러한 정보시스템의 다양한 수요와 복잡한 개발과정을 지원하기 위해 많은 도구들이 개발되었음에도 불구하고 여전히 생산성의 증가는 미비하며 개발비용은 증가하고 있는 실정이다. 또한 많은 투자에도 불구하고 여전히 실패 사례가 발견되고 있으며, 적기에 성공적으로 시스템의 개발이 완료되었다고 하더라도 개발된 시스템이 얼마나 조직의 목적에 기여하는가에 대해서는 의문이 제기되고 있다(Henderson & Lee, 1992).

최근 들어 인터넷이 보편화됨에 따라 많은 기업들이 새로운 e-Business 환경에 적응하기

위해 새로운 정보시스템 개발에 많은 비용과 시간을 투자하고 있다. Myer 등(1996)에 의하면 e-Business라는 새로운 패러다임의 출현으로 웹사이트를 단순히 정보 제공을 위한 외부 서버에 존재하는 문서의 집합이 아닌 쌍방향 의사소통을 제공하는 시스템으로 바라보는 시각이 확산되고 있다고 하였다. Piero(1999)는 기존의 정보 시스템 개발 개념과 의사소통을 위한 대중 매체 활용 개념을 동시에 고려한 인터넷 기반 응용 프로그램, 즉 웹개발 도구와 웹개발 접근법에 대한 많은 연구가 필요함을 지적하였다.

이러한 점을 고려하여 본 실험 연구에서는 웹사이트를 보다 역동적으로 만들어주는 프로그램 언어인 Perl/CGI와 PHP를 대상으로 생산성과 만족도에 대한 실험연구를 하였다.

1) 한국외국어대학교, 경영학과 교수, Tel : 02-961-4727, Email : ikchung@hananet.net
2) 한국외국어대학교, 경영학과 박사과정 수료, Email : leemm@chollian.net
3) Autoever 시스템관리부, Email : undersky01@korea.com

2. 웹언어

2.1 HTML(Hyper Text Markup Language)

HTML은 1969년 IBM에서 표준화된 문서를 교환하기 위해 개발한 SGML(Structured Generalized Markup Language)로 7비트 ASCII 텍스트로만 되어있다. 따라서 HTML로 작성된 문서는 컴퓨터 기종에 관계없이 서로 교환을 할 수 있고 디코딩을 하여 볼 수 있다. 1991년 CERN연구소에서 HTML를 WWW(World Wide Web)의 문서를 작성하는 표준 형식으로 채택함에 따라 본격적으로 인터넷에 등장하게 되었다. HTML은 웹에서 손쉽게 하이퍼텍스트, 하이퍼미디어 기능을 갖는 문서를 만들어 제공할 수 있고 이식성이 우수하며 사용이 편리하다. 반면에 HTML2.0, 3.2와 같이 고정된 태그만을 제공하기 때문에 사용자가 페이지 배치도나 멀티 칼럼과 같은 다양한 레이아웃을 할 수 없고, 문서 자체가 구조화되어 있지 않아 효과적인 검색 및 재사용이 불편한 단점이 있다.

2.2 Perl/CGI와 PHP

Perl(Practical Extraction And Report Language)은 1986년 Larry Wall에 의해서 개발된 유닉스(UNIX)용 텍스트 처리 언어이다. Perl은 텍스트 파일을 많이 사용하는 시스템 관리자들이 즐겨 사용하는데 그 이유는 기존의 셸 스크립트와 C언어로는 처리가 힘들었던 부분을 효과적으로 처리할 수 있게 해주기 때문이다. 특히 웹에서 Perl은 그 유용성이 매우 뛰어난데 그 이유는 회원등록, 주문서 등과 같은 여러 형태의 문서 형제가 웹에 등장하면서 이들을 처리해주는 프로그램, 즉 CGI(Common Gateway Interface)를 만드는 데에 Perl의 문자열 처리 능력이 우수하기 때문이다.

CGI(Common Gateway Interface)는 웹 페이지 제작 규약인 HTML정보의 단방향적인 단점에서 벗어나 웹사이트를 이용하는 사용자들이 정보를 주고받을 수 있도록 한 양방향적인 새로운 규약이다. CGI는 사용자가 웹 서버에 접속하면 웹 서버에서 CGI 프로그램을 호출하고 CGI프로그램은 그 실행결과를 다시 웹 서버에 전달하여 그 내용을 사용자가 받아보는 형식이다. CGI는 특별히 어떠한 언어로 제작해야 한다는 규칙은 없으나 일정한 웹 형식에 맞추어 C, C++, Perl 등과 같은 언어로 제작할 수 있다.

PHP(Professional HyperText Preprocessor)는 1994년 Rasmus Lerdorf에 의해 만들어졌으며, 몇 개의 매크로를 가진 파서(명령 해석) 엔진과 홈페이지 뒤에서 움직이는 몇 개의 유틸리티로 이루어져 있다. 1995년 파서가 재작성되어 PHP/FI Version 2가 개발되었고, FI는 Rasmus가 작성한 HTML 형식의 데이터를 해

석하는 별도의 패키지인데, Msql을 지원하도록 하여 PHP/FI를 만들어 냈다. PHP/FI는 1996년 후반 전세계적으로 15,000개 이상의 웹사이트에서 사용되었고 1997년 중반에는 사용하는 웹사이트가 50,000개 이상으로 늘어났다. 2001년 NETCRAFT의 통계 자료에 의하면, 전 세계의 300,000대 이상의 웹 서버에서 사용중이며, 도메인에서 사용중인 PHP까지 포함한다면 두 배 이상이 될 것이라고 추정하고 있다.

PHP(Professional HyperText Preprocessor)는 Server-side HTML-embedded Scripting 언어이다. C, Java, Perl 등에서 문장 형식을 빌려와서 사용하기 때문에 사용자가 배우기 쉬우며 빠른 시간에 프로그램을 만들 수 있다. 또한 웹브라우저 상에서 사용자가 실제 코드를 볼 수 없기 때문에 보안상 유리하다.

PHP를 CGI와 비교해 보면 PHP는 CGI보다 사용자에게 정보전달을 위한 단계가 간소하여 배우기가 쉬우며, 프로그램의 결과를 컴파일 없이 바로 사용이 가능하다. 그러나 PHP는 독립실행이 불가능하며, 웹에서만 사용 가능하다. PHP와 CGI를 차이점을 정리하면 다음 [표 1]과 같다.

[표 1] PHP와 Perl/CGI의 차이점 비교

PHP	Perl/CGI
독립실행 불가능	독립실행 가능
웹에서만 사용가능	웹이외의 콘솔 환경에서도 사용가능
웹서버와 일체형 - 서버부하가 적음	웹서버와 분리형 - 서버 부하가 많음
하나의 파일에서 HTML 사용 중 PHP의 코드작성이 가능함	HTML과 CGI프로그램은 각각 독립적이므로 CGI프로그램에서 HTML을 사용하려면 소스 코드에 강제로 HTML을 삽입해야 함
많은 데이터베이스를 쉽게 연동	데이터 베이스 연동이 어려움 (대부분 file system을 사용함)
배우기 쉬움	PHP에 비해 배우기 어려움
프로그램 결과를 컴파일 없이 바로 실행시킬 수 있음	각 CGI프로그래밍 언어마다 다름

3. 프로그램 언어에 관한 선행연구

Chrysler(1978)는 프로그래머의 생산성에 영향을 미치는 6가지의 변수 - 하드웨어(Hardware), 프로그래밍 모드(Programming Mode), 조직 특성(Organizational Characteristics), 프로그래머 특성(Programming Characteristics), 코딩 언어(Source Language), 응용시스템의 형태(Type of Application)- 을 개발하고 이들 6가지 변수들을 추정함으로서 프로그램 개발에 필요한 시간을 예상할 수 있는 틀을 개발하였다.

Welty & Stemple(1981)은 SQL과 TABLE에 관해 이를 사용하는 학생들을 대상으로 실험

조사를 하였다. 이 조사에서 복잡한 대화식 프로그램을 작성할 경우에는 절차적 언어를 습득한 학생들이 비절차적 언어를 습득한 학생들보다 성적이 좋았다. 그러나 Welty & Stemple의 연구는 대화식 단말기를 사용하지 않고 필답식으로 실험조사를 하였기 때문에 그 결과를 현대의 정보시스템 개발환경에 적용하기에는 무리가 있다.

Harel & McLean(1985)은 4세대 언어의 사용이 프로그래머의 생산성에 미치는 영향에 관해서 조사하였다. Harel & McLean은 Chrysler(1978)가 제시한 프로그래머 생산성 연구를 위한 연구체계로부터 6개의 독립변수(하드웨어 사양, 프로그래밍 모드, 조직특성, 프로그래머 특성, 코딩 언어, 응용시스템의 형태) 중에서 하드웨어 사양, 프로그래밍 모드, 조직 특성 변수를 상수(constant) 처리하고 나머지 3가지(프로그래머 특성, 코딩 언어, 응용시스템의 형태)를 독립변수로 선정하여 실험하였다. 종속변수는 프로그래머 생산성과 프로그램 효율성을 사용하였는데, 프로그래머의 생산성을 측정하기 위해 ① 프로그래밍 시간(programming time) ② 테스트 및 에러수정 시간(testing and debugging time)을, 프로그램 효율성 측정을 위해서 ① 프로그램 실행 시 CPU 처리시간(total CPU time for execution), ② 입출력 개수(total number of I/O operation), ③ 전체 코드 라인 수(total lines of source code)를 사용하였다.

연구결과는 프로그래머의 생산성 측면에서 4세대 언어가 3세대 언어에 비해서 매우 우수하다고 조사되었다. 효율성 측면에서는 4세대 언어가 3세대 언어에 비해서 CPU 처리시간은 느리지만 하드웨어의 성능향상과 가격하락, 그리고 소프트웨어 도구의 발전추이 등을 고려할 경우 이것은 큰 문제가 되지 않는다고 결론지었다.

Verner & Tate(1988)는 4세대 언어인 ALL을 사용하여 학사처리 정보시스템 개발 사례를 발표하였다. 이 연구에서 ALL을 사용하여 개발한 프로그램의 코드 라인 수를 Albrecht의 기능점 분석을 통해서 COBOL의 코드 라인 수로 환산하고 이를 다시 Boehm의 코코모(Cocomo)분석을 통해서 월간인력을 산출하여 실제로 개발에 소요된 월간인력과 비교하였다. 연구결과는 4세대 언어를 사용하는 것이 COBOL에 비해 4~5.3배의 노력을 감소시킬 수 있었다. 이러한 성과는 정보시스템의 복잡성에 따라 적절한 프로그램 언어를 사용하는 것이 합리적임을 말해준다.

Yen & Scamell(1993)은 두 가지 유형의 데이터 베이스 질의어 사용에 의한 사용자 성과와 사용자 만족의 차이에 관한 연구를 하였다. 일반적인 질의어인 SQL의 사용, 예제에 의한 질의어인 QBE(Query By Example)의 사용, 프로그래밍 경력, 그리고 질의어의 복잡성 등을 독립변수로 하고 이러한 요인들이 사용자 성과(작성된 질의어의 정확성, 질의어 작성 시간

과 에러 수정 시간)와 사용자 만족(사용 질의어의 구문 특성에 대한 만족, 해당 질의어 사용으로 산출된 인터페이스에 대한 만족)에 미치는 영향을 측정하였다. 연구결과는 모든 변수에서 QBE의 만족도가 높게 나타났다.

4. 프로젝트의 특성에 관한 연구

McFarlan(1981)은 개발 과업의 위험성을 조사하기 위하여 개발과업의 규모, 개발자의 경험, 개발과업의 구성 요소 등을 변수로 하여 연구하였다. 연구결과는 개발과업의 규모가 클수록 불확실성이 높아지며, 개발자의 경험이 높을수록 불확실성은 낮아지고, 과업의 구조화 정도가 높을수록 불확실성은 낮아진다고 하였다.

Nidumolu(1996)는 시스템 성과를 측정하기 위하여 사용자의 요구사항을 중심으로 연구를 진행하였다. Nidumolu가 사용자의 요구사항으로 연구의 범위를 제한한 이유는 소프트웨어 프로젝트가 진행되는 전체 과정에서 사용자의 요구사항이 매우 중요한 영향을 미치며 또한 시스템이 완료된 후에도 사용자의 요구사항은 중요한 역할을 하기 때문이라고 하였다.

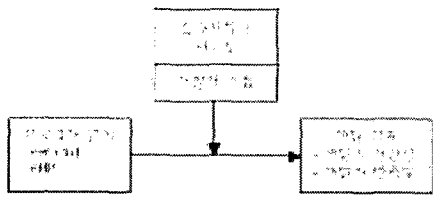
Rai & Al-Hindi(2000)는 시스템 실패의 원인이 기술적인 요소보다는 관리적인 요소에 있다고 보고 프로젝트 개발 모델과 위험성의 상호작용을 통해 프로세스의 성과 및 소프트웨어의 품질을 측정하고자 하였다. Rai & Al-Hindi의 연구에서도 위험성의 고려대상이 Nidumolu(1996)의 연구에서와 같이 사용자 요구사항이었으며, 주된 연구 대상인 프로젝트의 개발 모델링에 대해서는 관리자의 경험을 주요변수로 설정하여 측정하였다. 결과는 소프트웨어 프로젝트를 진행할 때에 일반적으로 한 가지 방법론에 근거해서 진행하기보다는 상황에 따라 관리자의 경험이 영향을 미치는 것으로 조사되었다.

5. 연구모형의 설계 및 가설의 설정

5.1 연구모형의 설정

본 연구의 목적은 프로그램 언어(Perl/CGI, PHP)가 웹 어플리케이션의 개발성과에 미치는 영향을 조사하기 위한 것이다. 프로그램 생산성에 관한 기존 연구와 해당 프로그램 언어의 특성에 근거하여 프로그램 언어(Perl/CGI, PHP)가 과업의 위험성에 따라 개발성과에 미치는 영향을 조사하기 위한 모델을 개발하였다.

과업의 위험성을 요구사항의 난이도와 개발자의 경험으로 구분하고, 프로그램 언어는 Perl/CGI와 PHP를 대상으로 하였다. 그리고 종속변수인 개발 성과를 측정하기 위해서 정보시스템의 개발의 생산성과 개발자의 만족도를 사용하였다.



[그림1] 연구모형

5.2 독립변수

본 연구에서의 독립변수는 Chrysler(1978)의 하드웨어 사양(Hardware), 프로그래밍 모드(Programming Mode), 조직특성(Organizational Characteristics) 등을 다음과 같은 환경 조성을 통해 처리하였다.

실험 대상자에게 각각 실험에 필요한 프로그램 명세서에 대해 명확히 설명하고 정해진 실험실에서 개발하되 실험 대상자들은 RAM 64 MB, CPU 펜티엄III 800으로 동일하게 구성된 22대의 컴퓨터를 사용하였다. 프로그래밍 모드(Programming Mode)와 관련하여 실험에서 수행되는 프로그래밍 작업은 각각 자신들의 개인용 컴퓨터에서 수행하도록 하였으며 작성된 프로그램은 중앙서버로부터 권한 부여된 계정을 통해 테스트 및 검사를 온라인으로 처리할 수 있도록 하였다. 조직특성(Organizational Characteristics)과 관련하여 실험에 참여하는 모든 대상자들은 Perl/CGI와 PHP에 대한 지식이 있는 사람들로 구성하였다.

5.3 조절변수

Harel & McLean(1985)의 연구에서는 Chrysler(1978)가 제시하였던 6가지 변수 중에서 3가지 변수를 독립변수로 활용하였다. 그러나 본 연구에서는 제한된 환경에서 세부적인 가설을 명확하게 규정하기 위하여 프로그램 언어만을 독립변수로 하고 나머지 두개의 변수는(요구사항의 난이도, 개발자의 경험) 다음과 같이 조절변수로 설정하였다.

요구사항의 난이도는 복잡한 유형과 간단한 유형에 대해 두개의 언어로 각각 작성하도록 하였다. 개발자의 경험과 관련해서는 업무경력이 1년 이상이고 실제 시스템 개발 경력이 있는 집단을 전문가집단으로, 전산전공 학생이거나 전산 관련 동호회에 활동중인 집단은 초보자집단으로 분류하였다.

5.4 종속변수

시스템의 개발성과를 측정하기 위해서 Harel & McLean(1985)의 연구에 근거하여 프로그램의 개발생산성과 McCracken(1978)과 Nidumolu(1996)의 연구를 바탕으로 개발자의 만족도를 종속변수로 선정하였다.

개발의 생산성은 Harel & McLean(1985)의 실험연구에서와 같이 설계시간, 코딩 시간, 그리고 프로그램 검사시간의 합으로 측정하였다. 이 경우 설계시간은 실험실의 제약성으로 인하여 각각의 측정은 생략하였고, 프로그램 작성시간으로 합하여 계산하였다. 개발 시간 측정은 두 집단에게 각각 두 유형의 프로그램 명세서와 최종 산출물의 인터페이스를 나누어 주고 작성하도록 하여 측정하였다.

개발자의 만족도는 실험자들이 개발에 사용하게 될 2가지 유형의 프로그램 언어(Perl/CGI, PHP)와 관련하여 각 언어에 대한 사용 용이성, 효율성, 확장성 등의 항목으로 구분하여 이를 설문지를 통해 작성하도록 하였다.

5.5 외생변수의 통제

본 연구의 목적은 서로 다른 프로그램 언어의 사용이 프로그래머의 생산성과 프로그램 언어 사용자의 만족도에 미치는 영향을 검증하는 것이다. 그러나 실제의 경우에 있어서 프로그래머 생산성에 영향을 미칠 수 있는 요소들은 기존의 연구에서 볼 수 있듯이 다른 외부 요인들이 많이 존재한다(Benbasat & Vessey, 1980; Nidumolu, 1996; Rai & Al-Hindi, 2000). 따라서 실험환경이 통제되고 여타 조건이 동일하도록 구성하기 위해 실험명세서의 설명을 자세히 하여 의사전달에 문제가 발생하지 않도록 하였다. 또한 프로그램 작성과 관련하여 개발도구의 사용에 대해서도 공통적으로 메모장을 사용하도록 하였다. 개발생산성은 설계, 작성, 검사로 구분하지 않고 프로그램이 작성되어 감독자로부터 확인을 받을 때까지의 시간으로 하였다.

5.6 연구 가설 설정

연구모형에 포함된 구성개념을 설명하기 위하여 다음과 같은 가설을 설정하였다.

- H1: 요구사항은 Perl/CGI와 PHP의 생산성에 유의한 영향을 미칠 것이다.
- H1.1: 간단한 요구사항에 대해서는 초보자의 경우에 PHP의 사용이 Perl/CGI 보다 개발생산성이 높을 것이다.
- H1.2: 복잡한 요구사항에서는 초보자의 경우에 PHP의 사용이 Perl/CGI 보다 개발생산성이 높을 것이다.
- H1.3: 간단한 요구사항에서는 전문가의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이다.
- H1.4: 복잡한 요구사항에서는 전문가의 경우에 PHP의 사용이 Perl/CGI 보다 개발생산성이 높을 것이다.
- H2: 개발자의 경험은 Perl/CGI와 PHP의 생산성에 유의한 영향을 미칠 것이다.

- H2.1: 간단한 요구사항에서는 Perl/CGI 전문가가 Perl/CGI 초보자보다 개발생산성이 높을 것이다.
- H2.2: 간단한 요구사항에서는 PHP전문가가 PHP 초보자보다 개발생산성이 높을 것이다.
- H2.3: 복잡한 요구사항에서는 Perl/CGI 전문가가 Perl/CGI 초보자보다 개발생산성이 높을 것이다.
- H2.4: 복잡한 요구사항에서는 PHP전문가가 PHP초보자보다 개발생산성이 높을 것이다.
- H3: PHP의 프로그램 만족도는 Perl/CGI의 프로그램 만족도보다 높을 것이다.
- H3.1: 간단한 요구사항에서는 PHP의 용이성이 Perl/CGI의 용이성보다 높을 것이다.
- H3.2: 복잡한 요구사항에서는 PHP의 용이성이 Perl/CGI의 용이성보다 높을 것이다.
- H3.3: 간단한 요구사항에서는 PHP의 효율성이 Perl/CGI의 효율성보다 높을 것이다.
- H3.4: 복잡한 요구사항에서는 PHP의 효율성이 Perl/CGI의 효율성보다 높을 것이다.

6. 가설의 검증

본 연구에서는 가설을 검증하기 위해서 이원배치 분산분석(two-way ANOVA)을 실시하여 개발의 성과에 영향을 줄 수 있는 개발자의 경험과 요구사항의 난이도 사이에 상호 작용 효과 존재 여부에 대해서 먼저 검증을 한 뒤, t-test에 의해 두 집단간의 개발생산성과 개발자의 만족도를 비교 분석하였다.

6.1 분산분석(two-way ANOVA)

두 요인 사이의 상호작용 효과를 보기 위한 이원배치 분산분석(two-way ANOVA) 결과는 [표 2] 와 같다.

[표 2] 분산분석(two-way ANOVA)

요구 사항	경험	소스	자유도	평균 제곱	F	유의 확률
간단함	-	경험/개발 언어	1	4.856E-03	.000	.987
복잡함	-	경험/개발 언어	1	5.814	.067	.797
-	초보자	요구사항/개발언어	1	208.000	3.881	.055
-	전문가	요구사항/개발언어	1	53.458	3.161	.085

[표 2]에서 볼 수 있듯이 유의확률이 0.05보다 크므로, 두 요인의 상호작용 효과는 없다고 볼 수 있다. 상호작용이 존재하지 않으므로 각 요인의 효과를 따로 분리해서 분석하는 것이 가능해진다.

6.2 요구사항의 난이도

개발자의 요구사항 난이도는 집단간의 비교를 통해 각 집단에 따른 생산성에 차이에 대해 t-test를 통해 검증하였다.

[표 3] 요구사항의 난이도

요구사항	경험	내용	t값	자유도	유의확률
간단한	초보자	개발언어/개발시간	-19.928	25	.000
	전문가	개발언어/개발시간	-13.450	35	.000
복잡함	초보자	개발언어/개발시간	-30.501	25	.000
	전문가	개발언어/개발시간	-17.376	35	.000

[표 3]에서 나타나듯이 검정통계량의 유의확률이 유의수준 0.05보다 모두 작다. 그러므로 유의수준 5% 하에서 요구사항의 유형에 따른 각 사용 언어 집단별 평균개발시간에는 차이가 있다고 결론을 내릴 수 있다. 즉 PHP와 Perl/CGI의 활용에 따른 생산성에 차이가 있음을 알 수 있다. 이를 토대로 초보자와 전문가 집단의 두 가지 언어 사용에 따른 차이점에 대한 가설을 검증하면 다음과 같다.

초보자 집단의 경우에 Perl/CGI와 PHP는 모두 작성시간에서 차이를 보였기 때문에 간단한 요구사항에 대해서는 초보자의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이라는 가설 H1.1과 복잡한 요구사항에서는 초보자의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이라는 가설 H1.2는 채택된다. 또한 전문가 집단에서는 Perl/CGI를 활용하여 프로그램을 작성할 때보다 PHP를 사용하여 프로그램을 작성할 경우 보다 높은 생산성을 보여주고 있다. 따라서 간단한 요구사항에서는 전문가의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이라는 가설 H1.3과 복잡한 요구사항에서 전문가의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이라는 가설 H1.4가 모두 채택된다.

6.3 개발자 경험

가설 H2에서 사용한 개발자의 경험을 검증하기 위해서는 요구사항 유형에 따른 동일한 언어를 각 집단별로 비교하였고 그 결과는 [표 4]와 같다.

[표 4] 개발자의 경험

요구사항	언어	종속변수	평균의 동일성에 대한 t-test		
			t값	자유도	유의확률
간단함	Perl/CGI	개발시간	1.004	20	.327
	PHP	개발시간	1.426	20	.169
복잡함	Perl/CGI	개발시간	5.496	20	.000
	PHP	개발시간	5.747	20	.000

[표 4]에서 나타나듯이 검정통계량의 유의수준이 요구사항이 간단한 경우에는 0.05보다 크다. 그러므로 유의수준 5% 하에서 두 집단간에 차이가 없다고 결론을 내릴 수 있다. 결국 간단한 요구사항에서는 Perl/CGI 전문가가 Perl/CGI 초보자보다 개발생산성이 높을 것이라는 가설 H2.1과 간단한 요구사항에서 PHP전문가가 PHP초보자보다 개발생산성이 높을 것이라는 가설 H2.2는 모두 기각된다.

반면에 요구사항이 복잡한 경우에는 유의수준 0.05보다 유의확률이 작기 때문에 복잡한 요구사항에서 Perl/CGI 전문가가 Perl/CGI 초보자보다 개발생산성이 높을 것이라는 가설 H2.3과 복잡한 요구사항에서는 PHP 전문가가 PHP 초보자보다 개발생산성이 높을 것이라는 가설 H2.4는 모두 채택된다.

6.4 개발자의 언어 사용 만족도

가설 H3을 검증하기 위해 용이성과 효율성의 두 가지 변수를 가지고 평균 비교를 통한 대응표본 t-test를 수행하였는데, 그 결과는 [표 5]와 같다.

[표 5] 개발자의 언어 사용 만족도

만족도	경험	요구사항	유의확률	평균비교
용이성	초보자	간단함	.002	Perl/CGI<PHP
		복잡함	.000	Perl/CGI<PHP
	전문가	간단함	.004	Perl/CGI<PHP
		복잡함	.000	Perl/CGI<PHP
효율성	초보자	간단함	.000	Perl/CGI>PHP
		복잡함	.000	Perl/CGI>PHP
	전문가	간단함	.001	Perl/CGI>PHP
		복잡함	.001	Perl/CGI>PHP

이를 통해 용이성에 대해서 초보자와 전문가가는 간단한 문제에 대한 언어비교 결과 검정통계량의 유의확률이 유의수준 0.05보다 작다. 그러므로 유의수준 5%하에서 개발자들의 두 가지 언어에 대한 용이성에 차이가 있다는 결론을 내릴 수 있다. 따라서 간단한 요구사항에서 PHP의 용이성이 Perl/CGI보다 높을 것이라는 가설 H3.1은 채택된다. 또한 복잡한 문제에 대한 유의확률 역시 유의수준보다 작기 때문에 복잡한 요구사항에서 PHP의 용이성이 Perl/CGI의 용이성보다 높을 것이라는 가설 H3.2 역시 채택된다.

효율성의 경우 초보자와 전문가 집단은 간단한 문제에 대해 검정통계량의 유의확률이 유의수준보다 작기 때문에 간단한 요구사항에

서 PHP의 효율성이 Perl/CGI의 효율성보다 높을 것이라는 가설 H3.3은 기각된다. 반면에 복잡한 요구사항에서 PHP의 효율성이 Perl/CGI의 효율성보다 높을 것이라는 가설 H3.4은 채택된다.

7. 결론

[표 6]은 본 연구에서 분석한 가설검증을 요약한 내용이다. 사용 언어에 따른 집단별(초보자집단/전문가집단) 성과의 차이에서는 요구사항이 간단할 경우에 두 개의 프로그램 언어 모두에 대해 초보자집단과 전문가집단 사이에 개발성과의 차이점을 발견할 수 없었다. 그러나 요구사항이 복잡할 경우에는 두 개의 프로그램 언어 모두에 대해 전문가 집단이 초보자 집단보다 개발성과가 높게 나타났다. 또한 사용 언어별 성과에는 PHP를 사용하는 경우가 Perl/CGI를 사용하는 경우보다 개발자 경험이나 요구사항의 난이도와 관계없이 높은 개발 성과를 보여 주었다.

[표 6] 가설 검증 결과

가설		구분	
H1	H1.1	간단한 요구사항에 대해서는 초보자의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이다.	채택
	H1.2	복잡한 요구사항에서는 초보자의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이다.	채택
	H1.3	간단한 요구사항에서 전문가 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이다.	채택
	H1.4	복잡한 요구사항에서는 전문가의 경우에 PHP의 사용이 Perl/CGI보다 개발생산성이 높을 것이다.	채택
H2	H2.1	간단한 요구사항에서는 Perl/CGI 전문가가 Perl/CGI 초보자보다 개발생산성이 높을 것이다.	기각
	H2.2	간단한 요구사항에서 PHP 전문가가 PHP 초보자보다 개발생산성이 높을 것이다.	기각
	H2.3	복잡한 요구사항에서 Perl/CGI 전문가가 Perl/CGI 초보자보다 개발생산성이 높을 것이다.	채택
	H2.4	복잡한 요구사항에서 PHP 전문가가 PHP 초보자보다 개발생산성이 높을 것이다.	채택
H3	H3.1	간단한 요구사항에서 PHP의 용이성이 Perl/CGI의 용이성보다 높을 것이다.	채택
	H3.2	복잡한 요구사항에서 PHP의 용이성이 Perl/CGI의 용이성보다 높을 것이다.	채택
	H3.3	간단한 요구사항에서 PHP의 효율성이 Perl/CGI의 효율성보다 높을 것이다.	기각
	H3.4	복잡한 요구사항에서 PHP의 효율성이 Perl/CGI의 효율성보다 높을 것이다.	채택

본 연구에서는 프로그램 언어의 선택과 과업의 위험성에 따라 개발성과에 차이가 있음

대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일-4일

을 보여주고 있다. 특히 보다 개선된 언어는
과업의 위험성을 감소시켜 줄 수 있다는 사실
이 발견되었다. 이러한 연구결과는 프로그램
언어에 관한 앞으로의 연구와 웹상에서 역동
적인 웹사이트 개발을 계획할 경우에 참고 자
료로 활용 가능할 것이다.

참고문헌

- Benbasat, I. & Vessey, I., "Programmer and Analyst Time/Cost Estimation", *MIS Quarterly*, Vol. 4, No. 2, June 1980, pp. 31-43.
- Chrysler, E., "Some Basic Determinants of Computer Programming Productivity", *Communications of the ACM*, Vol. 21, No. 6, June 1978, pp. 472-483.
- Harel, E. C. & McLean, E. R., "The Effects of Using Nonprocedural Language on Programmer Productivity", *MIS Quarterly*, June 1985, pp. 109-120.
- Henderson, J. C., & Lee, S., "Managing I/S Design Teams: a Control Theories Perspective", *Management Science*, Vol. 38, No. 6, 1992, pp. 757-777.
- Nidumolu, S. R., "Standardization, Requirements Uncertainty and Software Project Formance", *Information and Management*, Vol. 31, 1996, pp. 135-150.
- McCracken, D. D., "The Changing Face of Application Programming", *Datamation*, Vol. 24, No. 12, November 15, 1978, pp. 25-30.
- McFarlan, F. W., "Portfolio Approach to Information System", *Harvard Business Review*, Vol. 59, No. 5, September-October 1981, pp. 143.
- Piero, F., "Tools and Approaches for Developing Data-Intensive Web Application : A Survey", *ACM Computing Surveys*, Vol. 31, No. 3, September 1999.
- Rai, A. & Al-Hindi, H., "The Effects of Development Process Modeling and Task Uncertainty on Development Quality Performance", *Information and Management*, Vol. 37, 2000, pp. 335-346.
- Verner, J. & Tate, G., "Estimating Size and Effect in 4GL Development", *IEEE Software*, Vol. 5, No. 4, July 1988, pp.15-22.
- Welty, C. A. & Stemple, D., "Human Factors Comparison of a Procedural and a Non-Procedural Query Language", *ACM Transactions on Database Systems*, Vol. 6, No. 4, December 1981, pp. 626-649.
- Wrigley, C.D. & Dexter, A.S., "A Model for Measuring Information System Size", *MIS Quarterly*, June 1991, pp. 245-257.
- Yen, M. Y. & Scamell, R. W., "Human Factors Experimental Comparison of SQL and QBE", *IEEE Transactions on Software Engineering*, Vol. 19, No. 4, April 1993, pp. 390-409.