

□ 특별기고 □

컴퓨터 프로그래머 직무 분석

— 국내 전산인력 선발을 위한 제언 —

호서대학교 김 명 소
삼성생명 강 혜 련

● 목

1. 서 론
2. 직무분석(Job Analysis) 절차
 - 2.1 인터뷰 및 설문지 작성
 - 2.2 설문지 조사(Survey)
3. 직무분석결과

● 차

- 3.1 단순프로그래머와 시스템 디자이너의
최도점수 비교
- 3.2 선발차원(Selection Dimension) 도출
- 3.3 직무 기술서(Job Description)
4. 결론 및 논의

1. 서 론

정보화 시대를 맞이하여 컴퓨터 테크놀러지는 매우 다양한 분야에 응용되고 있으며, 이의 발달과 함께 인간의 일상생활양식도 커다란 변화를 가져오게 되었다. 최근 국내 컴퓨터산업이 빠른 속도로 확대됨에 따라 각 기업들의 전산전문인에 대한 수요가 급증하고 있다. 그러나 지금까지 제너럴리스트를 주로 양성해 온 우리사회는 전산 직종의 독특한 직무 특성을 반영하는 효과적인 인재선발 및 훈련을 실시하지 못하였기 때문에 다른 직종에 비해 이직률이 높은 결과가 야기되었던 것이다. 결국 현시점에서 우리나라가 우수 소프트웨어 개발국으로 성장하기 위해 가장 시급한 것은 전산 인력의 확충이며 필요한 잠재 능력을 갖춘 사람들을 선별해 낼 수 있는 효과적인 인재선발방법이라고 생각한다. 효과적인 선발을 위해서는 컴퓨터 프로그래머라는 직업에 요구되는 특수한 적성에 초점을 두어야 하며, 이와 무관한 특성들은 선발결정에 유리하게도 불리하게도 영향을 끼쳐서는 안된다[1].

그렇다면, 컴퓨터 프로그래머로 적합한 사람들은 어떤 특성들을 갖추어야 할까? 이에 대한

답변은 컴퓨터 프로그래머에 대한 체계적인 직무분석을 통해 직무의 본질을 이해 하는데서 출발한다. 직무분석이란 효과적인 인재선발을 위해 가장 기초가 되는 작업으로써, 적성검사와 같은 인사선발 도구가 타당성을 갖추기 위하여 필수적인 절차임에도 불구하고, 국내에는 물론 국외에서도 컴퓨터 프로그래머에 대한 직무분석 연구가 매우 부족한 실정이다. 따라서 본 연구는 컴퓨터 프로그래머가 직무를 성공적으로 수행하는데 요구되는 인적속성들(human attributes), 즉 지식, 기술, 능력 및 성격특성들을 도출해내기 위해 체계적인 직무분석을 시도하였다. 여기서 사용되는 분석방법은 작업자 지향적(worker-oriented) 접근법인데 이 방법은 직무수행에 요구되는 작업자의 행위 그 자체에 초점을 맞추기 때문에 우리가 궁극적으로 알고자 하는 인적속성(즉, 지원자의 자격요건)의 도출을 용이하게 해주는 이점이 있어 컴퓨터 프로그래머 적성검사 개발에 매우 유용하게 사용될 것이다[2].

2. 직무분석(Job Analysis) 절차

2.1 인터뷰 및 설문지 작성

직무분석의 첫단계는 전산관련 서적들을 참고로 기초적인 정보를 수집한 후 컴퓨터 프로그래머들과의 인터뷰를 통해 구체적인 업무 관련 정보를 얻는 것이다. 본격적인 인터뷰에 들어가기 앞서 하루동안 실제 컴퓨터 프로그래머 업무의 수행과정을 관찰하였고, 아울러 회사의 중간 관리자, 대학의 컴퓨터 관련 교수들과의 대화를 통해 직무에 사용되는 중요한 용어들을 인식하였으며 인터뷰시 행해질 질문들에 대한 아이디어를 얻었다.

본 연구의 인터뷰 대상은 대기업 전산실(삼성 데이터시스템), 컴퓨터관련 중소기업(삼보, 한국 MJL), 증권회사 전산실(동서증권), 대학부설 전산연구소(서울대 전산원)등에 근무하는 20명의 프로그래머들이었다. 이 과정에서 직무 수행시 중요하고 빈번하게 발생하는 업무들을 자유롭게 이야기하도록 하였고, 이를 토대로 백여개의 핵심적 직무구성요소들로 이루어진 업무(task) 리스트가 만들어졌다. 이 업무 리스트를 검토한 결과 각 업무의 수행에 필요한 인적 요구특성들(지식, 기술, 능력 및 성격특성들)을 대략적으로 이해할 수 있었다. 직무 수행자들이 일반적으로 공유할 수 있는 업무 개념과 인적속성들을 확인하기 위하여 업무 리스트를 바탕으로 지식, 기술, 능력, 성격 특성을 나타내는 80개의 문항들을 작성하였다.

인터뷰에 참가했던 프로그래머들 중 9명에게 80개의 문항들로 구성된 인적속성 리스트를 배포한 후 각각의 속성이 성공적인 업무 수행에 얼마나 필요한 가를 검토하도록 하였다. 9명중 3명이상이 필요하지 않다고 한 문항들은 제외하고 또 중복된 내용이 담긴 2~3개의 문항들을 함께 묶었다. 이 과정에서 컴퓨터관련 전문인들과 몇 차례의 brainstorming을 실시하여 신뢰성 있는 60문항으로 구성된 최종 설문지를 완성할 수 있었다. 이때 기본적 전제는 최종 선택된 60개의 인적 요구 특성들이 서로 중복되지 않고 동시에 프로그래머 업무 영역을 충분히 커버해야 한다는 것이었다. 또한, 위 특성들이 컴퓨터 업무와 타업무를 구별시키고 우수한 컴퓨터 프로그래머와 그렇지 못한 프로그래머를 변별하는 두가지 기능을 모두 갖추도록 노력하였다.

2.2 설문지 조사(Survey)

본 연구의 설문지는 세 가지 유형의 척도, 즉 사용 빈도 척도(Frequency Scale), 중요성 척도(Importance Scale), 입사전 요구 척도(Entry Requirement Scale)로 구성되어 있다. 우선 빈도 척도는 각 직무 요소들이 직무 수행에 얼마나 자주 사용되는가의 정도를 측정하는 것이고, 중요성 척도는 각 직무 요소들이 직무 수행에 얼마나 중요한 가를 보는 것이며, 입사전 요구 척도는 입사전에 지원자가 각 특성들을 예비적으로 얼마나 갖추어야 하는가의 정도를 평가한다. 각 척도는 Likert 유형의 7점 평정 척도이며, 만약 전혀 사용되지 않거나, 전혀 중요하지 않거나, 또 전혀 요구되지 않는 특성이라면 0으로 표시하게 하였다. 사용 빈도척도의 샘플을 표 1에 제시하였다.

전체 컴퓨터 프로그래머 집단을 대표할 수 있는 표본을 표집하기 위해 설문 대상자는 삼성 데이터시스템, 현대전자, 대우통신, 한국통신, 나눔 정보, 코아 기술, 삼보 컴퓨터, 한국 MJL, 동서증권, 서울대 전산원 등 다양한 회사및 연구소에 근무하는 200여명의 프로그래머들을 선택하였다. 컴퓨터 프로그래머의 업무는 시간이 흐름에 따라 계속해서 변화되고 업무의 특성이 세분화되어 가기 때문에 본 조사에서는 전체 컴퓨터 프로그래머 표본을 자신들이 체크한 직명(job title)에 따라, 코딩을 주로하는 단순직 프로그래머 집단과 디자인 혹은 디자인과 코딩을 함께하는 시스템 디자이너(혹은 시스템 어널리스트) 집단으로 구분하였다. 전체 200명중 162명(98명이 프로그래머, 64명이 디자이너)이 설문지에 응답하여 그 자료가 통계분석되었다.

3. 직무분석결과

3.1 단순프로그래머와 시스템 디자이너의 척도점수 비교

전반적으로 볼 때, 각 문항에 대해 평정된 값들은 빈도, 중요도, 입사전 요구 정도들에서 차이가 적었으며, 대부분 중간(4.0) 이상의 값들을

표 1 사용빈도 척도의 샘플

아래에는 컴퓨터 프로그래머의 업무를 효과적으로 수행하기 위하여 필요하리라고 여겨지는 지식, 기술, 능력 및 성격 특성들이 나열되어 있습니다. 나열된 대부분의 특성들이 사용되겠지만, 귀하가 현재 맡은 업무에 초점을 두고 생각해 볼 때, 어떤 특성은 상대적으로 더 자주 사용될 것이고 어떤 특성은 상대적으로 덜 쓰일 것입니다. 아래에 제시된 문항들이 귀하의 직무 수행에 얼마나 자주 사용되는 것인지를 판단하셔서 해당되는 번호에 ○표 해 주십시오.									
	전혀 사용하지 않는다	매우 드물게 사용한다			보통으로 사용한다			매우 자주 사용한다	
8	0	1	2	3	4	5	6	7	
실제된 서류를 보고 프로그램 언어를 사용하여 코딩하는 기술									
10.	0	1	2	3	4	5	6	7	
급변하는 첨단기술에 적응하는 능력									
17.	0	1	2	3	4	5	6	7	
실제시 일어날 수 있는 상황을 빠짐없이 확인하는 치밀성									
18.	0	1	2	3	4	5	6	7	
타인이 생각하지 못하는 아이디어를 산출해낼 수 있는 창의력									
25.	0	1	2	3	4	5	6	7	
컴퓨터 하드웨어(환경)에 관한 지식									

갖는 것으로 나타났다. 예비조사에서 각 문항들이 컴퓨터 관련 전문가들에 의해 1차적으로 그 필요성이 검토되었기 때문에 중간 이상의 평균값이 산출되리라고 어느 정도 예상되었었다. 또한 이러한 결과는 각각의 척도에 0점으로 표시된 ‘전혀 사용하지 않는다’, ‘전혀 중요하지 않다’, ‘전혀 요구되지 않는다’라는 보기에 응답한 빈도수가 매우 적게 나타난 결과(162명 중 1~4명 정도)와도 일맥상통한다.

각 문항의 표준편차를 살펴보면 대체적으로 1.5 미만으로서, 이는 응답자들간 의견차이가 크지 않았음을 반영해 준다. 그러나 ‘시스템 프로그래밍에 관한 지식’, ‘다루는 정보에 대해 오용하지 않는 도덕성’, ‘사용자에게 시스템을 설치해 줄 수 있는 기술’ 문항들은 표준편차가 약간 높았지만 평균은 4.33, 4.54, 3.79으로 나타나 문항을 그대로 유지하였다. 또한 각 척도의 신뢰도를 측정하기 위해 α 계수를 구해 본 결과 사용빈도 척도는 .9576, 중요성 척도는 .9553, 입사전 요구 척도는 .9746으로 문항들간 높은 내적 일관성을 보였다. 따라서 본 설문지(인적속성 척도)는 컴퓨터 프로그래머의 업무를 수행하는데 꼭 필요한 인적속성들을 포함하고 있어 타당성이

표 2 세가지 척도의 평균값.

	전체	프로그래머	시스템 디자이너
사용빈도	4.53	4.33	4.83
중요도	4.88	4.76	5.07
입사전 요구도	4.60	4.51	4.74

높다고 결론 지을 수 있었다.

전체 표본 및 프로그래머와 시스템 디자이너 집단에 대한 세 가지 척도의 평균값들이 표 2에 나타나 있다. 표 2을 살펴보면 각 척도의 평균들이 프로그래머 집단에서 보다 시스템 디자이너 집단에서 더 높았음을 볼 수 있다. 이 평균 차이를 t검증해 본 결과 세 척도 모두에서 통계적으로 유의미한 차이가 있는 것으로 나타났다. 이는 아마도 설문지에 제시된 지식, 기술, 능력 및 성격 특성들이 단순 프로그램을 짜는 업무와 시스템을 디자인하는 업무 모두를 포함하고 있기 때문인 것으로 생각된다. 시스템 디자이너의 영역이 시스템만 설계하는 중간 관리자급의 업무만을 담당할 수도 있고, 코딩업무를 겸하는 디자이너도

표 3 척도별 가장 높은 값을 갖는 다섯개의 문항들

	사용 빈도 척도	중요성 척도	입사시 요구 척도
전체	1) 소프트웨어 디자인이나 코딩시 논리적으로 사고할 수 있는 능력(5.25) 2) 개발업무에 대한 지식(5.18) 3) 프로그램 에러를 파악하고 수정하는 기술 혹은 능력(5.15) 4) 맡은 일을 정확히 기한 내에 끝내는 책임감(5.13) 5) 팀원을 성실껏 도울 수 있는 협동성(5.06)	1) 개발업무에 대한 지식(5.75) 2) 문제의 핵심을 정확하게 파악할 수 있는 통찰력(5.52) 3) 맡은 일을 정확히 기한 내에 끝내는 책임감(5.51) 4) 프로그램을 유지, 보수하는 기술 또는 능력(5.43) 5) 문제를 전체적 안목으로 볼 수 있는 종합적 사고력(5.39)	1) 맡은 일을 정확하게 마감시간을 지키는 책임감(5.51) 2) 실패해도 쉽게 포기하지 않는 끈기(5.45) 3) 힘든 상황을 참고 견디서 최종 결과를 지켜보는 인내력(5.38) 4) 팀원을 성실성의 것 도와 줄 수 있는 협동성(5.33) 5) 원만한 대인관계를 유지하는 능력(5.28)
프로그래머	1) 맡은 일을 정확히 기한 내에 끝내는 책임감(5.10) 2) 팀원을 성실껏 도울 수 있는 협동성(5.04) 3) 프로그램 에러를 파악하고 수정하는 기술 혹은 능력(5.03) 4) 프로그램을 유지, 보수하는 기술 또는 능력(4.96) 5) 개발업무에 대한 지식(4.96)	1) 개발업무에 대한 지식(5.65) 2) 프로그램을 유지, 보수하는 기술 또는 능력(5.48) 3) 맡은 일을 정확히 기한 내에 끝내는 책임감(5.37) 4) 목표의식이 강해서 실패해도 쉽게 포기하지 않는 끈기(5.34) 5) 어렵고 힘든 상황을 참고 견디는 인내력(5.32)	1) 맡은 일을 정확히 기한 내에 끝내는 책임감(5.10) 2) 팀원을 성실껏 도울 수 있는 협동성(5.04) 3) 프로그램 에러를 파악하고 수정하는 기술 혹은 능력(5.03) 4) 프로그램을 유지, 보수하는 기술 또는 능력(4.96) 5) 개발업무에 대한 지식(4.96)
디자이너	1) 디자인이나 코딩시 논리적으로 사고할 수 있는 능력(5.80) 2) 개발 업무에 관한 지식(5.52) 3) 문제의 핵심을 정확하게 파악할 수 있는 통찰력(5.36) 4) 목표의식이 강해서 실패해도 쉽게 포기하지 않는 끈기(5.34) 5) 프로그램 에러를 파악하고 수정하는 기술 혹은 능력(5.33)	1) 개발 업무에 관한 지식(5.91) 2) 문제의 핵심을 정확하게 파악할 수 있는 통찰력(5.78) 3) 맡은 일을 정확히 기한 내에 끝내는 책임감(5.72) 4) 문제를 전체적 안목으로 볼 수 있는 종합적 사고력(5.39) 5) 설계서 빠짐 없이 확인하는 치밀성(5.62)	1) 맡은 일을 정확하게 마감시간 내에 끝내려는 책임감(5.64) 2) 실패해도 쉽게 포기하지 않는 끈기(5.54) 3) 창의적인 아이디어 산출 능력(5.54) 4) 문제의 핵심을 정확하게 파악할 수 있는 통찰력 (5.45) 5) 힘든 상황을 참고 견디서 최종 결과를 지켜보는 인내력(5.44)

있으므로 단순 프로그래머 영역보다 광범위하다 하겠다. 따라서 프로그래머들이 자신들과 관련된 속성들만을 각 척도에서 높게 평정하고 디자이너와 관련된 일은 낮게 평정한 반면에, 시스템 디자이너들은 두 가지 업무에서 요구되는 속성들에 모두 높게 평정한 것으로 추측된다.

한편, 설문지에 포함된 지식, 기술, 능력 및 성격특성이 컴퓨터 프로그래머직에서 차지하는 상대적 비중을 살펴보기 위해 각 문항들의 점수를 세 가지 척도상에서 합친후 평균값을 구해 보았다. 그 결과 전체 표본에서는 책임감이 평균 5.38로 가장 높았으며, 끈기(5.29), 협동성(5.23)이 그 다음으로 높게 나타났다. 여기서 흥미로운 점은 가장 비중이 높은 3가지 속성들이 모두 성격특성이었다는 점이었다. 가장 낮은 점수로는

시스템 설치 기술(3.83)과 시스템 개발에 필요한 제반요구 사항을 추정하는 능력(4.12)이었다.

이를 프로그래머와 디자이너로 구분해서 살펴 보면 프로그래머들의 경우는 책임감(5.30), 협동성(5.20), 끈기(5.17), 인내력(5.14), 대인관계 능력(5.12)의 순으로 높게 나타나 전체집단의 경우와 유사했고, 낮은 점수를 갖는 특성들에는 위에서 나타난 것들 외에 통솔력(3.90)이 더 첨가되었을 뿐이었다. 반면 디자이너들의 경우는 논리력(5.53), 통찰력(5.53), 책임감(5.51), 끈기(5.47), 창의력 등의 순으로 높게 나타나 앞의 두 경우와는 조금 다른 양상을 보이고 있었다. 즉, 그 값의 차이가 크지는 않지만, 프로그래머들의 경우보다도 논리력, 통찰력, 창의력이 비중을 더 많이 차지하고 있었다. 디자이너 집단에서 낮은 평균

을 보인 인적속성들로는 변인을 기억하는 능력(4.08), 시스템 설치 기술(4.10), 키보드 조작 기술(4.15) 등이었다.

다음으로는, 각 인적속성의 비중도를 좀 더 구체적으로 검토, 비교해보기 위해 척도 별로 가장 높은 값을 갖는 문항들을 5개씩 나열해 보았다(표 3).

위의 내용을 간략히 요약한다면, 개발업무에 관한 지식, 책임감, 끈기 등은 두 집단 모두에게 중요하고 빈번히 사용되는 것으로 나타났지만 프로그램 에러를 수정하는 기술 (또는 능력), 프로그램을 유지 보수하는 능력들은 프로그래머 집단에게 더 중요하고 빈번히 사용되는 반면, 디자이너 집단에게는 통찰력, 논리적 사고, 종합력, 창의력 등이 더 중요하고 자주 사용하는 것으로 나타났다. 그러나, 여기서 주의해야 할 점은 대부분의 속성들이 평균값 이상의 높은 점수를 보였기 때문에 그 점수들에 대해서는 상대적 순위로만 평가하여야 하며 중요도나 빈도의 절대 점수가 낮다고 해석해서는 안 된다.

3.2 선발차원(Selection Dimension) 도출

다음으로 컴퓨터 프로그래머직의 수행에 요구되는 인적속성들의 기본구조 혹은 차원들을 이해하고 밝혀보기 위해 자료구조분석법 중 가장 대표적인 방법인 요인분석을 실시하였다. 요인분석 결과로 얻어지는 요인들 혹은 선발 차원(selection dimension)들은 선발도구 개발에 매우 유용하게 사용될 것인데, 그 이유는 컴퓨터 프로그래머 적성검사란 바로 이 요인들을 조작적으로 측정하는 도구이기 때문이다. 먼저 기술 및 능력특성에 해당하는 39개 문항들을 요인분석한 결과 10개의 선발차원들을 도출해 낼 수 있었으며, 각 차원 혹은 요인의 의미와 그 요인을 가장 잘 표현하는 대표적 문항들을 표 4에 나열하였다.

다음으로 인적속성 중 지식 특성에 관한 11개 문항들을 요인분석한 결과 3개의 차원이 도출되었다. 각 요인에 대한 해석 및 이에 속하는 문항들이 표 5에 나타나 있다. 특히 아래에 나열된 지식들은 기술, 능력, 성격과는 달리 테크니컬한

표 4 10개 요인의 의미와 대표적 기술 및 능력 특성 문항들

요인 1. 분석, 판단, 평가 능력

- 시스템의 필요성을 분석하고 시스템간 비교, 평가하는 능력
- 시스템 디자인에 쓰이는 모델링/시뮬레이션 기법들을 평가하는 능력
- 소프트웨어 프로그램 진단 능력
- 소프트웨어 프로그램 구성요소에 대한 판단 능력
- 소프트웨어의 기능, 목적, 제한점을 규정하는 능력

요인 2. 논리적 사고 능력 및 통찰력

- 디자이너나 코딩시 논리적으로 사고하는 능력
- 프로그램을 모듈로 분리해서 조직적으로 전개하는 능력
- 문제의 핵심을 정확히 파악할 수 있는 능력

요인 3. 테크니컬한 기술

- 설계된 다큐먼트를 보고 프로그램언어를 사용하여 코딩하는 기술
- 시스템을 설치해주는 기술
- 에디터 사용기술
- 키보드를 정확하고 신속하게 다루는 기술

요인 4: 표현 능력

- 아웃풋을 산뜻하고 보기 좋게 표현할 수 있는 미적 감각
- 소프트웨어 디자인을 도식화 시키는 기술 혹은 능력
- 소프트웨어 디자인 계획 및 최종보고서를 문서로 표현하는 능력

요인 5: 기획력 및 창의력

자신의 업무에서 새로운 일을 기획해내는 능력
타인이 생각하지 못하는 아이디어를 산출해내는 창의력

요인 6: 협동성

관련부서로부터 협력을 얻어내는 능력
구성원의 갈등, 역량, 성과를 인정하는 능력
팀원을 성실성의껏 돕는 협동성

요인 7: 수리적 능력

프로그램에 필요한 수리적 계산을 수행하는 능력
여러 가지 수리적 정보를 이해하여 수리적 판단을 하는 능력

요인 8: 적용 능력

급변하는 첨단기술에 적용하는 능력
프로그램 완성후 프로그램을 유지, 보수하는 기술 또는 능력

요인 9: 대인관계 능력

동료, 상사 및 사용자와 원만한 대인 관계를 유지하는 능력
업무내용을 알기쉽게 설명, 교육, 상담(컨설팅)하는 능력

요인 10: 스트레스 대처 능력

스트레스에 대처하는 능력
컴퓨터를 가지고 혼자서 지속적으로 업무 수행을 할 수 있는 독자적인 업무수행 능력

전문영역이었기 때문에, 이 요인들에 대한 해석은 컴퓨터 공학과 교수의 도움을 받아 이루어졌다.

마지막으로 10개의 성격 특성들을 요인분석하여 4개의 요인들을 추출하였다. 각 요인에 대한 해석 및 이에 속하는 성격특성문항들이 표 6에 제시되었다.

성격특성 요인들은 앞의 두 경우와 달리 요인 구조에 대해 명쾌한 결론을 내리기 어려웠다. 그 이유는 성격특성 문항들의 수가 적었고 각각의 문항이 하나의 요인을 나타낼 수 있는 일반적 특성들이었기 때문인 것 같다. 예를 들어 정서적 안정성은 하나의 중요한 요인으로 도출될 것이라 예측했는데 단일 문항이었으므로 끈기 혹은 탐구심과 함께 요인 1에 속하였다. 또한 여러 문항들이 능력 특성이 될 수도 있고, 성격 특성이 될 수도 있어서 분류상 어려운 점이 있었다. 예를 들면, 협동성, 독창성, 대인관계 특성들이 본 연구에서는 능력 및 기술 특성으로 간주되었고,

표 5 지식 특성 요인들의 의미와 각 요인에 속한 지식 특성 문항들

요인 1 : 프로그램 논리에 관한 지식
수리논리에 대한 지식
프로그래밍어에 대한 지식
자료구조에 대한 지식
알고리즘에 대한 지식

요인 2 : 프로그램 응용에 대한 지식
데이터 베이스에 대한 지식
소프트웨어 공학에 대한 지식
다양한 방면에 걸친 폭넓은 상식
개발업무에 대한 지식

요인 3 : 시스템 및 통신에 대한 지식
시스템 프로그래밍(예 OS)에 대한 지식
컴퓨터 통신에 대한 지식
컴퓨터 하드웨어(환경)에 대한 지식

표 6 성격특성 요인들의 의미와 이에 속한 성격특성 문항들

요인 1 : 끈기 및 정서적 안정성

실패해도 쉽게 포기하지 않는 끈기
 장시간 집중하는 집중성
 어렵고 힘든 상황을 참고 견디 끝까지 최종 결과를 지켜보는 인내력
 급변하는 컴퓨터 분야의 새로운 지식, 기술에 대한 탐구심
 정서적으로 안정된 차분함

요인 2 : 성실성 혹은 신뢰성

자신이 맡은 일은 어떻게 해서든지 정확히 마감시간을 맞추어 끝을 맺는 책임감
 다루고 있는 정보를 오용하지 않는 도덕성

요인 3 : 성취지향 및 소신성

남에게 인정받고 싶은 강한 성취욕구
 자신의 프로그램 작성 방법에 대한 소신 또는 주장성

요인 4 : 치밀성 혹은 꼼꼼함

설계시 일어날 수 있는 상황을 빠짐없이 확인하는 치밀성

의로 인해 표 6에 나타난 성격특성요인들은 종합적 혹은 포괄적이지는 못한 것 같다.

3.3 직무 기술서 (Job Description)

끝으로, 앞에서 제시된 세 척도상의 평균값들과 요인분석의 결과를 토대로 컴퓨터 프로그래머가 수행하는 직무내용을 체계적으로 설명해주는 직무기술서가 작성되었다.

4. 결론 및 논의

오늘날 조직의 성패는 유능한 인재의 확보에 달려있다고 해도 과언이 아니다. 유능한 인재란, 수행되는 직무와 독립적으로 생각될 수 없으며, 따라서 사람과 직무의 적합상태를 전제로 하게 된다. 이런 맥락에서 성공적 직무수행에 필요한 인적 요구사항을 밝혀주는 직무분석의 중요성은 아무리 강조되어도 지나치지 않은 것이다. 그러므로, 컴퓨터 프로그래머에 대한 본 연구의 직무분석 결과는 진산인력자원의 효율적 활용이라는 측면에서 몇 가지 시사점을 갖는다.

첫째, 컴퓨터 관련 분야의 인력 선발 체계가 아직 확립되지 않은 현 단계에서 볼 때 앞서 밝힌 직무분석 결과는 컴퓨터 프로그래머의 선발 도구를 개발하는데 기여할 것이다. 직무분석을 통해 성공적 직무수행에 요구되는 지식, 기술, 능력, 및 성격요인들이 도출되었으며, 이는 컴퓨터 프로그래머 적성검사 개발시 가장 중요한 자료로 사용된다. 현재 국내외적으로 사용되고 있는 컴퓨터 프로그래머 적성검사가 3~4종류밖에 되지 않고 또 그 대부분이 인지적 능력, 특히 학문적 지능(academic intelligence)만을 측정하고 있는 실정이다. 그런데 한가지 주지할 사항은 컴퓨터 분야가 관련능력이나 기술만큼 컴퓨터 마인드를 수용해 낼 수 있는 성격특성을 요구한다는 것이다. 따라서 본 연구결과를 통해 책임감이나 끈기 같은 성격적 특성과 아울러 실용적 지능(practical intelligence), 즉 창의성, 기획성, 상황판단력, 대인관계기술들이 매우 중요하다는 것이 입증되었으므로 이는 향후의 컴퓨터 프로그래머 선발도구개발에 반영되어야 하겠다.

둘째, 선발도구의 개발과 아울러 인적자원의 효율적인 활용은 사원교육 혹은 훈련(training)의

직무요약(Job Summary)

컴퓨터 프로그래머의 주된 책무는 시스템을 새롭게 설계하거나 기존의 시스템을 특정 목적에 맞게 변형시키는 것이다. 컴퓨터 프로그래머는 사용자 요구와 시스템 요구를 분석하여 목적을 설정한 후 프로그램 개발에 필요한 전반적인 계획과 논리를 세우고 이를 바탕으로 소프트웨어 설계, 즉, 도식화와 모듈링 작업을 수행한다. 또한, 프로그래머는 구체적으로 프로그램 언어를 사용하여 코딩한 후 테스트하여 에러를 검증한다. 마지막으로, 설계된 프로그램이 사용자 요구 및 개발 목적에 적합한지를 검토한 후 시스템을 유지 및 보수해 나간다.

직무기술서(Job Description)

직무내용	수행비중
1. 사용자의 요구분석 - 사용자의 요구를 정확히 알아낸다. - 사용자의 요구를 충족시킬 여러 대안들을 평가한다. - 기존 정보를 수집하여 최대한으로 활용한다.	10%
2. 개발 시스템(소프트웨어)의 요구분석 - 개발시스템의 특성(목적, 기능, 제한점)을 파악한다. 프로그램 조달방법의 여러 대안들(예: 구입, 개발, 혹은 기존 것 변환)을 평가한다. 시스템 개발에 따른 제반 비용을 추정한다.	10%
3. 시스템(소프트웨어) 설계 - 소프트웨어 설계 계획을 문서로 작성한다. - 시스템 설계에 적합한 모델링/시뮬레이션 기법을 선택하여 수행한다. - 프로그램을 모듈로 분리해서 전개시킨다. - 시스템 개발의 제반업무를 분장한다. - 소프트웨어 설계를 도식화시킨다.	25%
4. 프로그램 코딩 - 설계된 서류멘트를 보고 프로그램 언어를 사용하여 코딩한다.	18%
5. 테스트 - 프로그램 에러를 파악하고 수정한다. - 개발된 소프트웨어를 진단하기 위해 테스트 케이스를 만들어 본다. - 기존의 시스템과 개발된 시스템을 비교, 평가해본다.	12%
6. 최종 보고서 작성 - 개발된 소프트웨어에 대한 제반 보고 사항을 문서로 작성한다.	3%
7. 유지 및 보수 - 사용자의 필요에 따라 개발된 프로그램을 계속 유지 및 보수해 나간다.	10%
8. 커뮤니케이션 - 유관부서로 부터 소프트웨어 개발에 필요한 협조를 얻어낸다. - 개발된 소프트웨어 내용을 사용자에게 설명, 교육 및 상담(consulting)한다.	10%
9.기타 (*직무내용은 8번을 제외하고는 시스템개발 과정 단계 순으로 나열되었다.)	20%

중요성을 강조하고 있다. 직무분석 결과는 컴퓨터 프로그래머직의 성공적 수행에 필요한 구체적 과업(task)이나 작업 단계를 밝혀주고 있으므로, 훈련 내용이나 교과목을 포함한 체계적인 훈련 계획수립에 큰 도움을 줄 것이다.

마지막으로, 컴퓨터 프로그래머 직종에 관한 체계적인 연구가 국내에서는 아직 이루어지지 않았음을 감안할 때, 요인분석결과로 도출된 선발 차원들과 컴퓨터 프로그래머 직무기술서는 업무가 요구하는 특성 프로파일을 제시해주므로 컴퓨터관련 직업이나 학과를 선택하려는 예비 컴퓨터 프로그래머들의 진로상담(career counseling)에도 도움을 줄 수 있으리라 생각한다.

6. 참고문헌

- [1] Anastasi, A. Psychological Testing. New York: Macmillian, 1990.
- [2] Wernimont, P. Recruitment, Selection, and Placement. In S. Gael, The Job Analysis Handbook for Business, Industry, and Government. Volume II. New York: John Wiley & Sons, 1988.

김 명 소



1982 이화여자 대학교 영문학 학사
 1985 Univ. of Texas at Austin 상담심리학 석사
 1991 Univ. of Texas at Austin 측정심리학 박사
 1991 ~ 현재 호서대학교 산업심리학과 전임강사
 관심 분야: 심리측정 및 검사, 상담심리, 연구방법론

강 혜 련



1980 이화여자대학교 경영학 석사
 1983 Iowa State Univ. 경영학 석사
 1988 Iowa State Univ. 산업심리학 박사
 1988 ~ 현재 삼성생명 보험경영 연구소 선임연구원
 관심 분야: 인사 및 조직심리

● '94 인공지능연구회 춘계논문발표회 ●

- 일 자 : 1994년 5월 21일(토)
- 장 소 : 아주대학교 제 2이공관
- 주 관 : 인공지능연구회
- 문 의 : 아주대 김민구 교수

Tel (0331) 219-2437, Fax (0331) 213-8284