

ICT 기술자의 심리유형에 맞춰진 소프트웨어 개발 프로세스 교육 및 협업 능력 향상 방안

이재용¹

¹한서대학교 컴퓨터공학과

Software Development Process Improvement Training and Collaboration Capabilities Optimized to the Psychological Type of ICT Engineer

Jae-Yong Lee¹

¹Department of Computer Engineering, Hanseo University

요약 대부분의 소프트웨어 개발은 프로젝트 참여자들의 하드 스킬 능력 중심으로 판단되고 관리되고 운영된다. 그러나 현재의 소프트웨어 개발의 실패율은 매우 높다. 심리학의 발전으로 인간에 대한 탐험이 계속되면서 성격유형의 선호도가 업무의 선호도와도 연관이 있다는 연구가 활발히 이루어졌다. 심리유형론의 검사도구인 MBTI의 선호도는 ICT기술자들의 소프트웨어 개발주기에 맞춰진 업무의 선호도와 상관이 있음을 알게 되었다. 심리유형론과 소프트웨어 개발을 위한 소프트 스킬의 연관정보를 활용하면 소프트웨어 개발의 선호 공정을 이해할 수 있다. 본 논문에서는 ICT기술자들의 교육과 협업능력 향상 방안을 찾을 수 있도록 소프트웨어 개발 공정을 소프트스킬을 이용하여 성격유형별 분류가 가능하도록 프로그램을 개발하였다.

• **Key Words** : MBTI, 선호도, zig-zag, 문제해결, 소프트웨어 개발주기

Abstract Most of the software development is determined by the hard skills of project participants and capacity management and operations. However, failure of the current software development is very high. Since the development of the exploration of human psychology continues, the study that there is an association between the preferences of personality and work preference has been actively conducted. It is found out that there is a relationship of preferences in MBTI, the test tool of psychological typology, and those for the work of the ICT engineers tuned to the software development cycle. By using the information on the soft skills associated with software development for the psychological typology, it can be understood the preference of the software development process. In this paper, we develop a program for software development process to allow personality type to be classified with using the soft skills to find ways to improve education and collaboration capabilities of ICT engineers.

• **Key Words** : MBTI, preference, zig-zag, problem-solving, Software Development Cycle

*교신저자 : 이재용(dovejlee@gmail.com)

접수일 2015년 6월 30일

수정일 2015년 8월 10일

게재확정일 2015년 8월 20일

1. 서론

소프트웨어 개발은 기본적으로 주문형서비스로써 요구자의 주문을 충실하게 달성하면서도 요구자가 이해하지 못하는 기술적 영역에 대한 방법론을 선택하고 요구자와의 적절한 협력과정을 통하여 구현된다[1].

그럼에도 불구하고 ICT 기술자들의 교육과 능력평가가 소프트웨어 개발 능력에 중점을 두거나 개발방법론들의 하드 스킬(Hard Skill)에 초점을 두어 이루어져왔다. 또 주문자와 개발자들의 협업 부분에 필요성을 인지한다고 하더라도 리더십이라는 자기개발 영역의 과제로 바라보았다[2,3,4].

그러나 최근의 심리학의 급격한 발전으로 인하여 개발팀 협업 능력과 개발 요청자와의 적절한 의사소통능력인 소프트 스킬(Soft Skill)의 개발을 통해서 프로젝트를 성공으로 끌러는 노력이 계속되고 있다. 이러한 노력은 크게 두가지로 나눌 수 있는데 <심리적 기능에 따른 개인별 성향과 능력이 팀의 조직력을 좌우한다>는 관점이고 다른 하나는 <개인의 성격유형에 따라 소프트웨어 개발과정의 주요 서비스 공정의 선호도가 다르고 그 선호도는 소프트웨어 개발 프로세스와의 상관성이 높다>는 관점에 대한 연구결과들이 최근 보고되고 있다 [5,6,7,8,9].

이 소프트 스킬에 관한 두가지 관점은 모두 소프트웨어 개발 공정을 능률적 관점에서만 바라보는 기술자들의 생각과 관리자들이 관리의 편이성을 강조하기 쉬운 ICT 환경적 특이성 속에서, 소프트웨어 개발자들을 이해시키고 협업에서 활용되도록 하기에는 많은 어려움이 있다. 즉, 자기 이해를 기본으로 하는 심리학적 내용을 이해시키고 현장에서 활용할 수 있도록 하면 현재의 낮은 소프트웨어 개발 성공률을 높일 수 있는 방안이 될 수 있다. 그러나 이를 단편적 교육만으로는 개인의 심리적 요소들을 이해하게 위해서 그 효과를 기대하기가 어렵다. 성공적인 프로젝트의 수행이라는 거시적 파라다임으로의 사고 전환을 이끄는 데 어려움이 있기 때문이다.

본 논문에서는 소프트웨어 개발 팀 빌딩을 위한 조직심리학적 접근 방법론을 제시하고, 소프트웨어 개발자들이 쉽게 접근할 수 있는 자기 이해 및 성격유형별 소프트웨어 선호 공정의 이해를 위한 과정을 개발하였다.

2. MBTI와 ICT기술

2.1 MBTI

마이어스-브릭스 유형 지표(Myers-Briggs Type Indicator, MBTI)는 16개의 성격유형으로 구분된다. 이 검사는 캐서린 쿡 브릭스(Katharine C. Briggs)와 그의 딸 이사벨 브릭스 마이어스(Isabel B. Myers)가 칼 융(Carl Jung)의 심리유형론을 근간으로 1962년 개발하여 지금은 전세계 2억 5천 만명 이상의 사람들에 의해서 널리 사용되고 있는 심리검사이다[10].

최근 이 검사도구를 소프트웨어 개발자들에게 적용하기 위한 연구들이 진행되어 왔다. 그 연구는 크게 <일반인과 소프트웨어 기술자들의 유형 분포도의 차이에 대한 연구>와 <소프트웨어 개발 단계와 선호유형과의 상관연구>로 구분할 수 있다.

2.2 일반인과 ICT 기술자들의 성격유형 분포 연구

우리나라 성인의 성격유형분포도와 소프트웨어 기술자들의 성격유형 분포도의 비교연구는 소프트웨어 기술이 가지는 업무의 특성이 어떠한 성격유형의 사람들이 선호하게 되는지를 설명해 줄 수 있는 매우 유용한 자료이다. <Table 1>과 <Table 2>는 우리나라 성인 유형 분포표와 소프트웨어 기술자 성격 유형 분포표이다. 두 연구는 연구의 시점이 다르고 확률표집으로 이루어진 연구가 아니며 상관연구의 성격을 가지지 않으므로 절대적인 비교는 불가능할 것이다. 다만 ENFP를 제외하고는 감정

<Table 1> Adult Personality Type Distribution in KOREA

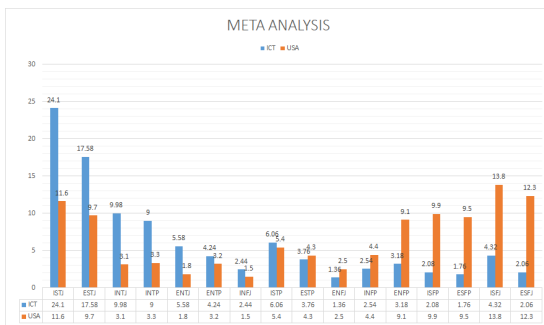
ISTJ 21.37%	ISFJ 8.17%	INFJ 2.44%	INTJ 5.48%
ISTP 7.88%	ISFP 6.34%	INFP 3.73%	INTP 3.33%
ESTP 5.39%	ESFP 5.38%	ENFP 3.07%	ENTP 2.20%
ESTJ 14.09%	ESFJ 5.60%	ENFJ 1.87%	ENTJ 3.46%

<Table 2> ICT Engineer Personality Type Distribution in KOREA

ISTJ 22%	ISFJ 4.2%	INFJ 1.4%	INTJ 5.6%
ISTP 8.1%	ISFP 3.8%	INFP 1.4%	INTP 4.7%
ESTP 6.6%	ESFP 4.2%	ENFP 4.7%	ENTP 3.8%
ESTJ 20.7%	ESFJ 4.2%	ENFJ 1.4%	ENTJ 3.3%

(Table 3) Soft-skills At Software development

communication skill
interpersonal skill
Ability to work independently
Active listener
strong analytical and problem solving skills
open and adaptable to changes
innovative
organization skill
pay thorough and acute attention to details
fast learner
Team player



[Fig. 1] Meta-analysis of 5 Research for ICT Engineer and U.S.A. Adult

(Feeling)을 의사 결정수단으로 하는 모든 유형의 분포 비율이 상대적으로 적은 것의 유의미한 것 있는지에 대한 추가 연구가 필요할 것이다[10,11].

[Fig. 1]은 미국의 ICT 기술자들의 성격유형을 연구한 5개의 연구에 대한 문헌 연구(meta analysis)결과이다. 이 연구에서는 내적 선호도인 ST, NT의 경우 전체 인구에 비해 높은 분포를 보였다. NF, SF의 경우 상대적으로 낮은 분포도를 보였다. 이 메타연구의 결과는 심리적 내적 선호유형 ST이나 NT일수록 인구분포보다 많으며, SF이거나 NF일수록 더 적게 분포한다는 것을 의미한다 [6].

이는 심리적 선호(Preference) 기능이 소프트웨어 기술자들에게 주어진 개발환경과 상호작용하기 위한 선호도의 적절성을 의미하는 것이다.

2.3 소프트웨어 개발 주기와 MBTI 상관연구

소프트웨어 개발 단계는 분석-디자인-프로그래밍-테스트-유지보수 과정으로 나뉘어진다. 이 개발 프로세스에 적합한 성격유형에 대한 연구는 ICT 기술자들의 진

로상담과 프로젝트 생산성 향상에 크게 기여할 수 있다. 이 연구는 5개의 개발 프로세스에서 요구되는 각각의 업무 요구사항을 소프트 스킬과 연관하고 각 소프트 스킬에 적절한 MBTI 성격유형을 찾는 방식으로 작업에 적합한 선호 유형을 찾아나가는 연구 방법론이다[5,8].

즉, 각 소프트웨어 개발단계에서 요구되는 업무 요구사항을 정의하고 여기에 적합한 소프트스킬을 매칭 시키는 방법으로 연구가 이루어져 있다. 이 연구에서 사용하는 소프트 스킬의 목록은 <Table 3>과 같다[5,8].

2.3.1 소프트웨어 분석

시스템 분석 단계는 각 구성 요소의 높은 식별능력이 요구되며 소프트웨어 시스템의 분해 능력이 뛰어나야 한다. 이 단계에서는 시스템 분석가로서 사용자의 요구를 고려하여 시스템의 필수적인 기능을 이해하고, 이러한 요구 사항을 충족하는 추상 애플리케이션 모델을 만드는 능력이 필요하다.

소프트웨어 분석과정에서 필요한 기능 중에서 소프트 스킬과 일치하는 부분은 <대화기술>과 <대인관계 기술>로써 MBTI 유형으로는 외향(E)과 감정(F)의 기능이 주로 필요한 기능이다.

2.3.2 소프트웨어 설계

소프트웨어 설계자는 넓게 볼 수 있는 능력을 가져야 한다. 이를 위해서는 패턴을 분별하는 직관(N)을 필요로 한다. 즉, 많은 양의 자료 중에서 관련 항목을 정확하게 분리할 수 있어야 한다. 상상력을 지니고 혁신적인 성향을 가진 사람들로써, 설계에서 직관적 능력을 보여야 한다. 소프트웨어 설계자는 프로토타입을 포함하는 작업(task)과 처리 기능을 정교화하며 입력 및 출력을 정의하는 등 넓은 범위를 수행해야 한다. 이 단계에서는 주문자와 팀 내의 토론등의 상호 작용이 이루어져야 하며 분석 단계에서 필요한 것과 유사한 기술이 필요로 한다.

소프트웨어 디자인에서 적합한 여러 기능 중에 소프트 스킬과 매칭되는 것은 <강력한 분석 및 문제해결 능력>과 <혁신>이며 이에 필요한 MBTI 성격 유형은 직관(N)과 사고(T)의 기능이 주로 필요하다.

2.3.3 프로그래밍

프로그래밍은 프로그램의 디자인을 세련된 버전으로 만드는 것을 포함한다. 이 단계는 제어 구조, 관련 변수

및 데이터 구조의 식별, 프로그래밍 언어의 구문과 특성의 상세한 이해를 수반한다. 주로 위에서 아래로 작업하며, 반복 단계적으로 각 내용을 정제하는 과정을 따라야 한다.

이와 매칭되는 소프트스킬은 <독립적으로 작업할 수 있는 능력>, <강력한 분석 및 문제해결 능력>, <세부사항에 사고대처 및 정확한 주의> 따라서, 프로그래머는 세부 사항에 참석하고 논리적이고 분석적인 사고 스타일을 유지해야 한다. MBTI 유형으로는 내향(I), 감각(S)과 사고(T)의 기능이 주로 필요한 기능이다.

2.3.4 테스트

테스트 단계에서는 소프트웨어의 결함을 찾고 결함이 발견이 이루어지는 주요 단계이다. 이 단계의 주요 초점은 가능한 많은 결함을 찾아내는 것이다. 각 모듈을 다른 구성 요소들과 분리하여 개별적인 실험을 하는 테스트, 다양한 예상 입력과 모듈 기능에 적합한 통합 테스트, 검증 및 전체 소프트웨어가 제대로 작동하는지 확인하는 시스템 테스트 프로세스로 구분된다. 테스트 전략은 조직적이며 체계적으로 접근해야 한다.

결함이 발견 된 후, 디버깅 과정은 프로그래밍의 과정에서의 의사 결정을 조정, 변경하도록 소프트웨어 엔지니어를 이끌 수 있어야 한다.

이와 같은 기술들은 소프트 스킬 중에서 <조직화 기술>과 <세부사항에 사고대처 및 정확한 주의>에 해당되며 이는 MBTI 유형으로는 감각(S)과 판단(J)의 기능이 주로 필요한 기능이며, 테스트 단계에 매우 유용한 성격 기능이다.

2.3.5 유지 보수

소프트웨어는 완성 후에도 지속적으로 변경 될 수 있다. 뿐만 아니라, 운영하는 동안 시스템을 유지보수를 위한 내용이 끊임없이 변화하는 특성이 다. 따라서 유지보수의 원활함과 소프트웨어 시스템을 향상에 필요한 작업들은 실제 현실과 관찰력 경향이 더 감각(S)인 유형의 사람들이 맞아서 하는 경향이 있다.

모든 가능성을 탐구하고 변화와 적응에 더 개방적인 사람들이 유지 보수를 즐기며, 사용자가 요청하는 일정에 더 공감적으로 대응한다.

그런 사람들은 실제적인 문제를 해결하고 프로그램 및 시스템의 변화를 즐기기를 좋아하기 때문에 문제 해

결 능력과 실무 접근 방식은 유지 보수를 위해 훌륭한 자원이 된다. MBTI 유형으로는 감각(S)과 인식(P)의 기능이 주로 필요한 기능이다.

<세부사항에 사고대처 및 정확한 주의> <변화에 대한 개방성과 적응성> 의 소프트 스킬이 유지보수에 적합한 기술이다.

2.3.6 소프트웨어 수명 주기와 성격유형

위와 같이 각 단계와 가장 관련이 있는 성격유형에 따라 소프트웨어 라이프 사이클 모델의 다섯 가지 주요 단계를 도식할 수 있다. <Table 4>는 특정 성격 유형 더 영향을 미칠 수 있는지를 개념화한 프레임 워크이다. 이 모델에서 소프트웨어 수명주기 모델을 발생하는 단계를 시스템 분석, 설계, 프로그래밍, 테스트 및 유지 보수의 5 단계로 본다[6].

5 단계에 영향을 미친다는 성격 유형 배후 이론은 더 많은 소프트웨어 라이프 사이클의 일부 단계에 영향을 미칠 가능성이 있다는 것을 의미한다. 또 <Table 4>는 각 단계와 가장 관련이 나타나는 성격 유형을 보여준다. 이것은 하나의 단일한 성격유형을 가진 엔지니어가 모든 다양한 스펙트럼의 작업에 모두 어울리지는 않는다는 것을 인식할 수 있게 된다. 반대로 어떤 유형도 소프트웨어 개발주기 각각에서 매우 중요한 전문가가 된다는 것을 표현하는 것이다.

소프트웨어 프로젝트 작업을 할당 할 때, 각 기술자의 강점을 고려하는 것이 유리하다는 것을 말해준다. 즉, 소프트웨어 공학은 성격 유형의 다양성을 필요로 한다. 기술과 성격 특성의 다양성은 소프트웨어 개발 및 유지 보수와 관련된 수많은 문제를 해결할 수 있으며, 다양한 성격유형을 보유한 조직은 팀의 구성에 있어서 강한 팀을

<Table 4> Personality Type and Software life cycle stages

Personality Type	software life cycle stages				
	Analysis	Design	Programming	Testing	Maintenance
E	✓				
I			✓		
S			✓	✓	✓
N		✓			
T		✓	✓		
F	✓				
J				✓	
P					✓

구축할 수 있을 것이다. 강력한 팀은 다양한 관점을 가지고 있기 때문에 그들의 소프트웨어 엔지니어의 스타일이나 개성을 다양화하기 위한 의식적인 시도할 수 있을 것이다. 즉, 프로그래밍 심리학을 통한 소프트웨어 공정의 이해와 자기 탐색, 팀 빌딩은 다양성을 통한 능률적 협업을 이끌 수 있게 됨을 말해주고 있다.

또, 소프트웨어의 개발 공정은 팀원 전체가 진행되는 정신과정이며 심리유형론을 주장한 칼 구스타프 융의 자기실현의 과정상의 진행과도 일치하는 것이다.

3. S/W 기술자의 전문성과 문제해결

3.1 소프트웨어 엔지니어의 전문적 자질

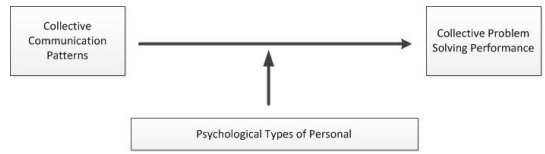
소프트웨어 개발자들은 개인적 습관이나 성격에 관한 문제의식을 가지지 못하고, 시스템 분석, 설계, 유지 보수 같은 소프트웨어 엔지니어링 절차에 몰입한다. 그 결과 개발 과정상에 적합한 특징 중심의 개인별 평가를 하지 못하고 프로그래밍에 몰두하게 되므로 다음과 같은 고도의 정형화된 행동과 생각들을 가지게 된다[3,13].

- 사회적 상호작용성이 매우 낮다.
- 도전과 성취에 대한 욕구가 너무 높다.
- 관리 책임에 대한 욕구가 낮다.
- 공식적으로 권위가 낮다.
- 대인관계 갈등에 대한 내성이 적다.
- 고용주보다는 직업에 대해 충성심이 높다.
- 시간관념이 최적화되어 있다.
- 문제 해결을 위한 조직적인 접근을 시도한다.
- 안정적인 보안 업무에 대한 관심이 높다.

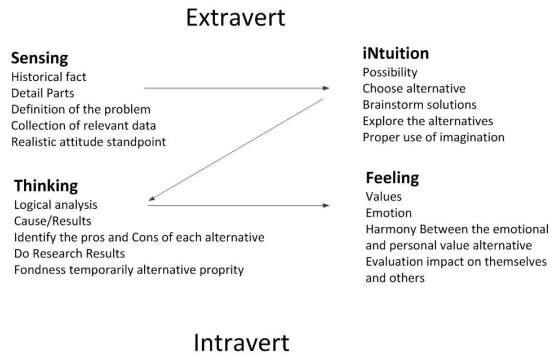
개발의 성공여부는 개발팀 협업 능력과 개발 요구자와의 적절한 의사소통 결과에 따라 좌우된다.

3.2 집단 문제 해결에 대한 수렴모델

ICT 기술자들은 조직 내부/외부의 매우 다양한 사람들을 만나 프로젝트를 진행하게 된다. 대부분의 기술자들은 집단 내부부의 의견수렴 절차의 진행의 어려움으로 프로젝트 실패율이 높다. 수렴에 도달하기 위해서 팀들과 동일화하거나 팀원의 지향성을 보완하도록 주의의 방향성을 설정해야 한다. [Fig. 2]에서 처럼 집단의 의견수



[Fig. 2] Collective problem-solving skills development & Psychological Type of Influence



[Fig. 3] Zig-Zag process at Problem Solving

렴 절차와의 괴리를 해결하는 방법론들이 제공되었다 [13,14,15].

이러한 심리유형의 영향력을 가지는 성격유형을 알면 팀의 구성의 효율성을 달성할 수 있을 뿐만 아니라, 팀 운영에서 갈등요소를 줄이고 ICT 프로젝트를 보다 성공적으로 이끌 수 있다. 지그제그(Zig-Zag) 프로세스는 요구설계과정 동안에 프로젝트 개발팀을 위해서 유용하다. [Fig. 3]에서는 이의 과정을 보이고 있다[6,14].

첫 단계에서는 프로젝트의 현재 사실들은 무엇이며, 설계 개발 비용은 얼마나 되고, 어떤 솔루션이 가장 적합한지를 확인하기 유리한 단계이다. 이전에 시도했던 일은 무엇이며 팀이 현재하고 있는 것은 무엇인가로 감각(S)에 맞는 작업 내용이다.

두 번째 단계에서 솔루션이 의미하는 다른 무엇이 존재하는지를 요약할 있다. 다른 방법이 없는지의 브레인 스톰(BrainStorm)을 통한 창조성에 접근해 볼 수 있다. 어떤 아이디어와 관련이 있는지의 호기심을 확인하며 기본이 되는 패턴이 무엇인지의 이론적 내용을 확인할 수 있으며 새로운 것에 조치가 취해지는 것이 무엇인지를 확인할 수 있다.

세 번째 단계에서 프로젝트의 장단점이 무엇인지를 논리적으로 확인할 수 있으며 이 솔루션의 더 필요한 것은 무엇이며 더 필요한 것이 무엇인지를 확인한다. 더불

어 이 솔루션이 잘못된 점이 무엇인지를 비판적으로 살펴보고, 올바른 대응법에 대하여 살펴보게 된다.

네 번째 단계에서 프로젝트에서 우리가 좋아하는 것이 무엇이며, 고객에 어떻게 고객에 영향을 미치는 지에 대해 배려하고, 어떻게 이해당사자들을 지원하고 즐겁게 하는지를 확인한다. 이에 이 솔루션의 올바른 점은 무엇인지, 팀 내에서 합의한 내용이 무엇인지를 확인한다.

4. 콘텐츠의 개발

4.1 프로그램의 구성

MBTI 성격 유형과 소프트웨어 개발프로세스의 이해를 통한 생산성을 향상시키기 위한 프로그램 개발을 위한 프로그램 내용은 다음과 같이 구성할 수 있다.

- 1) 교육프로그램에 대한 소개
- 2) 친밀감 형성작업
- 3) 교육에 관한 소개 및 검사실시
- 4) 4가지 지표에 대한 강의
- 5) 유형 해석지를 통한 자기 성격 유형 이해
- 6) 유형별 집단활동
- 7) 유형별 작업과 소프트웨어 개발 주기 이해
- 8) 유형별 개발단계의 이해 집단활동
- 9) 소프트웨어 개발단계와 zig-zag 모델

이 프로그램에서 일반적인 프로그램의 소개 및 교육 과정에 대한 강의 및 유형 작업을 통하여 MBTI 성격유형별 선호지표가 뚜렷하게 느낄 수 있도록 하는 것이 중요하다.

유형 작업을 통하여 충분히 선호지표에 대한 인식을 높인 후에 기술자들의 선호 유형과 프로그래밍의 상관성을 느낄 수 있는 유형 작업을 하거나, 리커드 척도 등을 이용한 문항 작업을 통하여 성격유형과 소프트웨어 라이프 사이클에 대한 선호를 확인 하는 유형 작업이 필요할 것이다.

소프트웨어 라이프 사이클에 따른 그룹작업의 모델들은 다양하게 구성할 수 있을 것이다. 이는 위하여 내향 - 외향의 그룹 작업을 다음과 같이 한다. 먼저 참여자들을 E 집단과 I 집단으로 나눈다. 그 후에 두 집단에게 프로그래밍 과정과 시스템 분석과정의 내용들 중에서 각각 하나의 작업을 요청한 결과를 확인하거나 작업 선호를

판단할 수 있도록 한다.

새로이 감각 - 직관의 그룹 작업을 위하여 참여자들을 감각(S) 집단과 직관(N) 집단으로 나눈다. 그 후에 두 집단에게 프로그래밍/테스트/유지보수 과정과 소프트웨어 디자인의 내용들 중에서 각각 하나의 작업을 요청한 결과를 확인하거나 작업 선호를 판단할 수 있도록 한다.

사고 - 감정의 그룹 작업을 위해서는 참여자들을 사고(T) 집단과 감정(F) 집단으로 나눈 후에 소프트웨어디자인/프로그래밍 작업과 시스템 분석과정의 내용들 중에 하나를 요청하여 그 결과를 확인하고 발표를 통하여 선호도가 얼마나 다른지를 판단할 수 있도록 한다.

판단 - 인식의 그룹 작업을 위해서는 판단(J)과 인식(P)의 두 그룹으로 나눈 후에 테스트과정과 유지보수 과정의 작업들 각각을 수행하게 하도록 한후 발표를 통하여 선호 작업의 차이점을 발견할 수 있도록 한다.

이와 같은 과정을 통하여 각 참여자는 선호 유형의 소프트웨어 개발 프로세스에 미치는 영향을 이해할 수 있게 되고, 각 공정에 대한 자신의 선호도를 깊이 인식할 수 있게 될 것이다. 이러한 과정은 궁극적으로는 낮은 프로젝트 성공률을 높일 수 있는 심리적 자원을 제공할 수 있을 것이다.

5. 결론

ICT 기술자들은 매우 물리적 환경에서 기계와의 작업을 통하여 인간과 컴퓨터와의 상호작용을 효과적으로 돕기 위한 역할을 담당한다. 그러나 주어진 환경은 매우 물리적인 상황이며 하드스킬에 의존하여 문제의 해결에 집중하여 결과물을 만들어 내게 된다. 이러한 환경은 낮은 소프트웨어 개발 성공률에 이르게 함에도 불구하고 각 프로젝트가 독립적으로 진행되는 이유로 이 문제에 접근하는 방법이 부족할 수밖에 없었다.

심리학의 발전은 이러한 문제에 대한 새로운 접근법을 제시하고 있다. 즉, 자신이 선호하는 유형의 작업이 무엇인가의 탐색을 통하여 소프트웨어 개발이라는 커다란 정신적 과정에 접근할 수 있을 뿐만 아니라, 자신이 접근하기 어렵거나 이해하지 못했을 수 있는 과정에 접근할 수 있는 새로운 관점을 제시하였다.

본 논문에서 제시하고 있는 MBTI 성격유형론을 통한 소프트웨어 개발 프로세스의 선호도의 이해 및 접근을 위한 방법론을 제시하였다. 이 방법론은 ICT 기술자들을

위한 워크샵을 진행할 수 있는 기본 자료로 활용될 수 있을 것이다.

앞으로 이 논문의 주요 작업내용을 이용한 척도개발을 하여야 할 것이다. 또, 많은 심리학자나 컴퓨터 공학자들은 태도기능(I/E, J/P)의 기능 보다는 심리기능(S/N, T/F)이 소프트웨어 개발 주기에 더 깊은 관련을 것이라는 가설에 대한 후추 연구가 필요할 것이다.

REFERENCES

- [1] Tracy L. Lewis, Wanda J. Smith, Building Software Engineering Teams That Work: The Impact of Dominance on Group Conflict and Performance Outcomes, 38th ASEE/IEEE Frontiers in Education Conference, S3H1 ~ S3H5, 2008.
- [2] Sang Cho, Blockly webc Programming Convergent Learning System, Journal of the Korea Convergence Society, Vol. 6, No. 1, pp. 23-28, 2015.
- [3] Robert N. Charette, "Why Software FAILS", IEEE Spectrum, pp. 42 - 49, 2005.
- [4] Tracy L. Lewis, Wanda J. Smith, "Creating High Performing Software Engineering Teams : The Impact of Problem Solving Style Dominance on Group Conflict and Performance", Journal of Computing Sciences in Colleges, Vol. 24, No. 2, pp. 121-129, 2008.
- [5] Luiz Fernando Capretz, Faheem Ahmed, "Making Sense of Software Development and Personality Types", IT Pro, Jan/Feb, pp. 6-13, 2010.
- [6] Rien Sach, Marian Petre, Helen Sharp, "The Use of MBTI in software engineering", In conjunction with 2010 IEEE Symposium on Visual Languages and Human-Centric Computing Universidad Carlos III de Madrid, pp. 21 - 25, 2010.
- [7] Luiz Fernando Capretz, "Clues on Software Engineers' Learning Styles", International Journal of Computing & Information Sciences, Vol. 4, No. 1, pp. 46 - 49, 2006.
- [8] Peter G. Drexel, Christian Roberson, "Using MBTI in a Project-Based Systems Analysis and Design Course", International Conference on Engineering and Meta-Engineering, 2010.
- [9] Luiz Fernando Capretz, "Personality types in software engineering", International Journal of Human-computer Studies, pp. 207-214, 2003.
- [10] Jeong-taek Kim, Hye-suk Sim, Seok-bong Je, MBTI development and utilization, 1988.
- [11] Hong, Ji Soo, A study on Job stress and Stress-Coping Stress-Coping Stress-Coping Method by Personality Types : the Cases of workers in Information Technology industry, SOOKMYUNG Women's University Graduate School of education, Master dissertation, 2007.
- [12] Rien Sach, Marian Petre, Helen Sharp, "The Use of MBTI in software engineering", PPIG 22nd Annual workshop, 2010.
- [13] J. S. Kam, S. Syed-Abdullah, A. J. Cowling, M. Holcombe, "A Study into the effects of personality type and methodology on cohesion in software engineering teams. Behaviour & Information Technology, Vol. 26, No. 2, pp. 99-111, 2007.
- [14] Myoung-jun Kim, In-sik Jeon, "Troubleshooting Using the MBTI", Korea Industrial and Organizational Psychology, 2002 Fall Conference and Symposium, pp. 89-99, 2002.
- [15] Nancy J. Barger, Linda K. Kirby, Changes in the organization and type, ESSESTA, 2002.

저자소개

이재용(Jae-Yong Lee)

[정회원]



- 1985년 2월 : 인하대학교 대학원 전자계산학과 (전산학석사)
 - 2002년 2월 : 인하대학교 대학원 전자계산공학과 (전자계산공학박사)
 - 2015년 2월 : HYCU 휴먼서비스 대학원 심리상담학과(문학석사)
 - 2000년 3월 ~ 현재 : 한서대학교 전자계산공학과 교수
- <관심분야> : 프로그래밍 심리학, 뇌컴퓨터 상호작용, 심리정보과학