

S/W개발자의 IT 프로젝트 참여 의도 연구: 조절초점에 따른 프로젝트 기술 최신성의 효과를 중심으로

S/W Developer's IT Project Participation: Focusing on the Moderating Role of Regulatory Focus on the Effect of Technology Recency on Participation Intention

박요한 (Yohan Park) NHN Comico Japan¹

박도형 (Do-Hyung Park) 국민대학교 경영대학/비즈니스IT 전문대학원²

Abstract

In the success of an IT project, the role of S/W developer in carrying out the project is critical and crucial. However, compared to its importance, there is only a few studies on the role of human resources in software engineering field, mainly focusing on technology. Based on regulatory focus theory, this study examines how the effect of technology recency on developer's intention to participate in the IT project changes depending on the regulatory focus of the software developer. As a result of the experiment, it was confirmed that there is a difference in the effect of technology recency on project participation intention according to the condition of developer's regulatory focus. This study is meaningful because it extends regulatory focus theory to software engineering field and made theoretical contribution to the research field of behavioral software engineering. In practical, it was possible to suggest a way to improve the intention of the developer to participate in the project, by understanding the software developer in the IT project, considering the software developer propensity and project characteristics.

Keywords: IT Project, Regulatory Focus, Behavioral Software Engineering, Software Developer, Technology Recency

1. 서론

IT 프로젝트는 본질적으로 다양한 성격을 가지고 있다. IT 프로젝트는 다양한 산업과 사업 기능을 지원하면서 그 적용범위가 넓고, 다양한 분야의 전문가들을

필요로 한다. 그에 따른 각 영역에 대한 요구기술이 다르고, 의사소통의 문제 등이 발생하기도 한다. 또한 새로운 기술이 생겨나고, 비즈니스 요구 변경에 따라 IT 프로젝트 관리의 빠른 대응이 요구되고 있다(Schwalbe 2015).

¹ park.yohan@nhn-comico.com

² 교신저자 dohyungpark@kookmin.ac.kr

IT 프로젝트와 소프트웨어 개발 프로젝트가 복잡성과 다양성을 가지고 있는 만큼, IT 프로젝트의 성공 또는 실패요인에 대해서 이해하고, 해당 요인들의 분류 체계 및 모델들을 만들기 위한 다양한 연구가 이루어졌다(Butler and Fitzgerald 2001; Lyytinen and Hirschheim 1987; McLeod and MacDonell 2011; Poulymenakou and Holmes 1996; Scott and Vessey 2002). IT 프로젝트의 성공을 결정 짓는 요인에는 환경, 조직, 프로세스 등 다양한 요소들이 있지만, 가장 중요한 요소로서 본 연구는 프로젝트를 수행하는 ‘사람’ 즉, 소프트웨어 개발자를 주목하고자 한다.

기존의 연구들이 성격, 개인의 행동, 인지와 같은 인간적 요소가 개인과 팀의 성과에 영향을 준다고 밝히고 있고(Bradley and Hebert 1997; Constantine 1995; Faraj and Sproull 2000; Guinan et al. 1997; Rasch and Tosi 1992), 성공적인 프로젝트들은 소프트웨어 실무자에 대한 이해와 동기부여가 소프트웨어 개발의 생산성 향상과 고객만족으로 이어져 프로젝트가 성공한다고 설명하고 있다(Procaccino et al. 2002; Procaccino et al. 2005).

이렇게 IT 프로젝트 성공에 있어, 프로젝트를 수행하는 ‘사람’이 중요함에도 불구하고, 소프트웨어 공학 분야에서의 ‘사람 Soft- or Human-related’과 관련된 연구가 부족한 실정이다(Lenberg et al. 2015). 이에 따라 최근 급부상하고 있는 ‘행동 소프트웨어 공학(Behavioral Software Engineering)’은 소프트웨어 공학에 인지분야, 행동분야, 사회과학 분야의 이론들을 접목시키려는 시도로 소프트웨어 공학에 속한 인간적 속성에 대한 이해를 목표로 한다고 볼 수 있다(Lenberg et al. 2015).

소프트웨어 개발자들의 동기와 특성 연구에 있어 현재까지는 주로 MBTI와 고전적 동기이론 중심으로만 연구가 진행되었다(Capretz 2003; Cruz et al. 2015; Feldt et al. 2010; Hall et al. 2009; Smith 1989; Tur-

ley and Bieman 1995). 일부 연구자들은 소프트웨어 공학 분야에서 이론의 사용과 실증 연구가 부족하다는 점을 지적하였고(Hannay et al. 2007), 소프트웨어 공학 분야 외, 다른 영역의 이론들을 적용한 연구의 필요성의 제안하기도 하였다(Kitchenham et al. 2004).

본 연구는 소프트웨어 개발자의 특성과 IT 프로젝트 간의 관계를 규명하고자 한다. 어떤 IT 프로젝트의 조건과 소프트웨어 개발자의 특성이, 소프트웨어 개발자의 IT 프로젝트 참여도에 영향을 미치는지를 알아보고자 한다. 본 연구에서는 IT 프로젝트의 조건으로 프로젝트에 ‘도입되는 기술의 최신성’을 설정하였으며, 소프트웨어 개발자의 특성 유형은 Higgins의 조절초점 이론(Regulatory Focus Theory)를 통해 개발자의 특성과 동기 유형을 설명하고자 한다. 이를 통해 ‘행동 소프트웨어 공학’ 분야의 이론적 확장에 기여하고, 소프트웨어 개발자 특성의 이해와 IT 프로젝트 성공 요인을 도출하여, 실무적 시사점을 얻고자 한다.

2. 이론적 배경 및 연구 가설

2.1 소프트웨어 개발자와 프로젝트 성과의 관계

McLeod and McDonell(2011)은 기존의 연구들을 종합하여 시스템 개발 프로젝트의 성과에 영향을 미치는 요인들에 대한 프레임워크를 제시하였다. 프로젝트 성과에 영향을 끼치는 차원으로 ‘Project Context’, ‘Development Process’, ‘Institutional Context’, ‘People and Action’을 제시하였다. 이 중 ‘People and Action’은 소프트웨어 시스템 개발 프로젝트와 관련되어 있는 개인과 그룹에 대한 내용이다. 프로젝트에 참여하는 사람들에 대한 특성과 행위, 상호작용과 관계들은 프로젝트의 진행과정과 프로젝트 성과를 다양한 방법으로 결정짓는다. 따라서 시스템 개발 과정 동안 사람들의 역할과 행동에 대한 이해가 필요하다.

1990년대 초반부터 시작된 여러 연구들이 성격, 개인의 행동, 인지와 같은 인간적 요소가 소프트웨어 공학 분야에서의 개인과 팀의 성과에 영향을 준다는 것을 밝히고 있다(Bradley and Hebert 1997; Constantine 1995; Faraj and Sproull 2000; Guinan et al. 1997; Rasch and Tosi 1992). 소프트웨어 개발자 관점에서 프로젝트의 성공을 살펴본 연구들에 의하면, 소프트웨어 실무자에 대한 이해와 동기부여가 소프트웨어 개발의 생산성 향상과 고객만족으로 이어져 프로젝트를 성공시킬 수 있다고 한다(Procaccino et al. 2002; Procaccino et al. 2005).

소프트웨어 개발자들은 소프트웨어 개발에 어떻게 접근하여 수행할지와 프로젝트 산출물을 만들어내는 과정에 영향을 줄 수 있는 특징들을 가지고 있다. 이러한 특징들은 기술, 역량, 전문성, 경험, 인간관계, 커뮤니케이션과 사회적 기술 등을 포함하고, 어플리케이션 영역의 지식, 헌신, 동기와 신뢰, 규범, 가치와 믿음을 포함한다. 개발자의 기술과 역량의 중요한 변화는 개발 생산성에 영향을 끼칠 수 있고, 나아가 프로젝트 결과에도 영향을 끼칠 수 있다(Fitzgerald 1998).

여러 연구에서 IT 개발 프로젝트의 성공은 프로젝트와 관련된 이해관계자 특히, 소프트웨어 개발자의 능력에 좌우한다고 보고하고 있다. 소프트웨어 개발자의 잠재적 능력을 발휘하기 위해서는 동기와 소프트웨어 개발자의 행복이 가장 기본이 되며, 일련의 연구들이 행복을 느끼고 동기가 유발된 팀 구성원들이 프로젝트 생산성과 결과 품질에 긍정적 영향을 준다는 것을 보여주고 있다(Beecham et al. 2008; Boehm 1988; Curtis et al. 1988).

2.2 소프트웨어 개발자의 동기

동기(Motivation)는 인간과 관련된 요인들 중 하나로, 관리하기 어려운 속성을 가지고 있지만 생산성과 품질, 프로젝트 실패와 관련이 있다는 사실 때문에 학

자들과 현업에서 주목을 받기 시작했다(Da Silva and França 2012). 동기는 사회과학에서 오랫동안 연구되어 온 분야이며, 동기 이론들은 조직의 이익을 위해 직원들의 생산성을 극대화하는데 초점을 맞추고 있다. 동기 이론들은 직원들을 활동적으로 관리할 수 있고, 인센티브가 주어지면 동기가 유발되지만, 그렇지 않을 경우 직원들이 업무를 덜 효과적으로 수행할 것이라는 전제를 가지고 있다. 동기 이론들은 일반적인 이론이기 때문에 소프트웨어 엔지니어의 활동에 관해서도 적용될 수 있다(Hall et al. 2009).

소프트웨어 공학에서 동기는 생산성과 소프트웨어 품질 관리에 가장 큰 영향을 주는 단일 요인이라고 보고되기도 하였다(Boehm 1981; McConnell 1998). 소프트웨어 개발자의 동기는 조직적 맥락과 기질, 프로젝트 팀의 문화로부터 영향을 받는 경향이 있다. 개발자들은 효과적인 리더십, 긍정적인 업무 환경, 소속감, 긍정적 피드백과 합리적 수준의 자율성 또는 책임을 통해 동기를 받을 수 있다(Oz and Sosik 2000).

Sharp et al.(2009)는 Beecham et al. (2008)의 연구를 바탕으로 체계적인 문헌 조사를 통해 새로운 모델을 제시했다. MOCC(Motivators, Outcomes, Characteristics, Context)라고 불리는 새로운 모델은 소프트웨어 엔지니어의 특징, 중개자, 조정자, 동기요인, 동기상실요인, 외부적 징후와 관련한, 소프트웨어 공학에서의 동기에 관한 종합적인 관점을 제시한다. MOCC 모델은 동기요인(Motivators), 성과(Outcome), 소프트웨어 엔지니어 특징(Software Engineer Characteristics), 맥락적 요인(Contextual Factors)까지 네 가지 요소들의 관계를 보여주고 있다.

그밖에 새로운 솔루션을 디자인하는 기술적 도전, 시스템 개발에 있어 새로운 기술이나 표준 방법론을 사용하는 기회, 최고 경영자와 가깝게 일하는 것, 또는 사용자를 이롭게 하는 것들 또한 일부 개발자에게는 동기유발이 될 수 있다(Fitzgerald 1998; Fitzgerald and

Fitzgerald 1999; Jones and Harrison 1996; Kautz 2004; Linberg 1999; Nandhakumar and Avison 1999). 한편, 개발자들은 자율성의 부족, 불충분한 개발 자원, 기술적으로 비현실적인 요구사항들, 팀원과 다른 관리지간의 갈등, 개발 경험의 부족, 직업 불완전성, 부적절하거나 강제적인 표준방법과 도구의 사용 등 다양한 이유들로 좌절하거나, 프로젝트 목표에 충분히 헌신하지 못할 수 있다(Linberg 1999; Nandhakumar and Avison 1999).

이상의 연구들을 볼 때, 소프트웨어 개발에 있어 개발자가 가지고 있는 개인적 특성들과 프로젝트에 대한 헌신이 프로젝트 성공과 성과에 영향을 미치는 중요한 요인이고, 이러한 개발자의 태도와 행동에 영향을 주는 내재적, 외재적 요인들이 다양하다는 사실을 알 수 있다.

개발자의 이해와 동기 요인이 프로젝트 성과에 중요한 역할을 하지만, 이에 대한 이론적 연구가 아직은 미흡하다. Hall et al.(2009) 은 1980년부터 2000년까지 발표된, 소프트웨어 공학에서 동기와 관련된 연구 92건에 대해 조사하였고, 어떠한 동기이론들을 사용하는지 분석하였다. 조사에 따르면 92개의 연구중에서 절반정도인 51건의 연구들 만이 고전적 동기이론을 사용하였고, 기타 이론을 활용한 연구는 15건 정도가 있었다. 이들은 소프트웨어 공학 분야에서 동기이론의 프레임워크와 기본적인 원리에 대한 이해와 적용이 부족하다고 주장하고 있고, 낮은 이론 사용은 개별 연구의 발견에 대한 설득력을 낮추게 되고, 지식 체계의 일반화를 더디게 한다고 지적하고 있다.

2.3 소프트웨어 개발자의 특성에 관한 연구

소프트웨어 개발자의 특성을 밝히는 연구에서, 개인의 성격유형을 밝히기 위해 지금까지 가장 많이 쓰이는 측정도구는 MBTI(Myers-Briggs Type Indicator)이다(Myers et al. 1998). MBTI를 이용한 연구에 따르면 소

프트웨어 엔지니어들은 분명히 사회의 평균적인 사람들보다 감정적이기 보다는 지나치게 이성적이다. 일반적으로 소프트웨어 엔지니어들은 감정이나, 인간적 측면에 기반한 결정보다는 객관적이고, 냉정하며, 분석적인 사고와 판단을 한다(Capretz 2003; Turley and Bieman 1995).

Capretz(2003) 는 100명의 소프트웨어 엔지니어를 대상으로 MBTI 측정을 하였는데, 일반인에 비해 소프트웨어 엔지니어들은 ‘직감-객관’ 유형과 ‘오감-객관’ 유형이 두드러지게 나타났지만, ‘오감-주관’과 ‘직감-주관’ 유형은 매우 낮게 나타났다. 이와 반대되는 연구도 있는데, Smith(1989)의 연구에서는 37명의 시스템 분석가의 35%가 ‘ISTJ(내향-오감-객관-계획)’형으로 나타났고, 30%는 ‘ESTJ(외향-오감-객관-계획)’형으로 나타났다. 그러나 MBTI가 사람들을 서로 다른 성격 유형으로 분류는 하지만, 서로 다른 성격차원 간의 강도를 고려하지 않는 한계가 있다. 이는 통계적인 분석을 어렵게 하여, 성격요인과 개발자의 의사결정 및 성과 사이의 관계를 파악할 수 없도록 한다(Feldt et al. 2010).

Couger and Zawacki(1980) 가 1000명 이상의 소프트웨어 전문가를 대상으로 한, 직업 만족도에 대한 연구에 따르면, 소프트웨어 전문가들은 대체로 다른 직업군 보다 높은 성장 욕구를 가지고 있다. 소프트웨어 전문가들은 높은 내적 동기요인을 가지고 있고, 성장에 대한 만족감, 높은 품질의 성과를 추구한다. 따라서 소프트웨어 전문가들은 자신의 성장 욕구와 직업의 잠재적 동기가 일치할 때에 동기 유발이 커진다. Beecham et al.(2008) 의 연구에서도 개발자들은 일반적으로 도전과 배움에 대한 욕구를 가지며, 성장 지향적이고, 독립을 추구하여 자율적으로 행동하기를 원한다고 한다.

2.4 조절초점이론

Higgins는 자기차이이론을 확장하여 인간의 동기와 정서를 설명하기 위해 조절초점 이론을 제안하였다

(Higgins 1998, 2000). 인간의 욕구, 이상과 당위, 행동의 결과로 나타는 성과의 획득 또는 비획득, 손실 또는 비손실의 조건과 같은 심리적 요인들이 향상 또는 예방이라는 자기조절 초점에 영향을 주고, 개인의 동기과 정서 그리고 행동전략에 영향을 준다고 주장 했다. 여기서 향상초점(Promotion Focus)의 동기는 쾌락을 추구하고 이익에 접근하려고하는 성향을 말하며, 예방초점(Prevention Focus) 동기는 고통을 미리 예방하고, 손실을 회피하려는 성향을 말한다(Higgins 1997).

향상초점의 사람들은 목표에 대한 성취, 향상, 증진, 열망과 같은 진취적 성향이 있어, 새로운 상황에 대한 도전적인 의식이 있다(Higgins et al. 1997). 이들은 손실과 비손실 여부보다는 이득을 얻거나 얻지 못하는 상황에 대해 더 민감해 한다(Higgins 1997; Lee et al. 2000). 그리고 성취하는 것에 민감 하기 때문에 최대한 목표를 성취하려고 하고, 이로 인한 변화를 받아들이는 성향과 위험을 추구하는 경향을 보이며, 이는 “모든 방법에 의한 발전 추구(Persue All Means of Advancement)” 특징이라고 할 수 있다(Brendl and Higgins 1996; Liberman et al. 1999). 반면 예방초점의 사람들은 실패하는 것에 민감하기 때문에 최소한의 목표를 추구하는 성향을 보이며, 향상초점 사람들과 달리 위험성 있는 판단을 하려는 성향이 약하고, 현재 상태를 유지하며 안정을 추구하려는 경향이 있다. 이는 “조심스럽게 어떤 실수도 피하는 것(Carefully Avoid Any Mistake)”라고 설명할 수 있다(Brendl and Higgins 1996; Crowe and Higgins 1997; Liberman et al. 1999).

또한 향상초점과 예방초점의 사람들은 목표달성에 따른 감정의 지각도 다르게 나타난다. 향상초점 성향의 사람들은 목표를 달성할 때에 즐거움을 느끼지만, 예방초점 성향의 사람들은 안도감을 느낀다. 만약 목표를 달성하지 못한다면, 향상초점 성향의 사람들은 낙담을 느끼지만, 예방초점 성향의 사람들은 불안의 감정을 지각한다(Camacho et al. 2003).

Higgins(2000)는 개인의 조절초점과 특정 목표를 추구하기 위한 수단이 일치 하는 경우, 그 수단에 대해 긍정적인 태도와 높은 가치를 둔다고 설명하였고, 이를 조절적합성(Regulatory Fit)이라고 설명하였다.

향상초점의 경우, 희망, 이상, 열망, 성취, 도전, 접근, 열망, 감정적과 같은 정보나 수단에 적합성이 나타나고, 이에 대해 높은 가치를 두어 평가한다. 반면에 예방초점은 의무, 안정, 책무, 책임감, 회피, 감시, 이성적과 같은 정보나 수단에 적합성이 나타나고, 이에 대해 높은 가치를 둔다. 이는 사람들이 자신의 조절초점과 일치되는 속성이나 정보를 접하거나, 조절초점과 일치하는 전략이나 활동에 참여할 때에 ‘옳다고 느끼는 상태(Feeling Right)’가 형성이 되고, 이런 느낌이 제시된 속성이나 정보, 참여하는 활동이나 수단에 긍정적인 영향을 미치기 때문이다(Higgins 2000).

이러한 연구결과들은 개인의 조절동기는 정보처리를 하는 개인의 기질적 성향에 따라 구별되기도 하나, 정보를 처리하는 맥락에 따라 달라지고, 어떠한 동기에 집중되었느냐에 따라 개인의 판단기준이 다르게 나타난다는 것이다. 즉, 조절 동기가 개인의 기질적 성향보다는 정보를 처리하거나, 선택을 해야하는 상황 및 대상에 따라 달라진다는 것을 보여준다(Crowe and Higgins 1997; Roese et al. 1999; Shah and Higgins 1997).

2.5 조절초점에 따른 프로젝트 도입기술의

최신성이 참여의도에 미치는 영향

IT 프로젝트에서 새로운 기술의 도입과 적용은 많은 위험성을 가지고 있다. 새로운 기술은 기존의 문제를 해결하고, 효율성있게 목표를 달성하게 도와주기도 하나, 새로운 기술의 도입은 기존에 알려지지 않은 새로운 문제를 만들어낼 가능성이 존재한다. 결국 프로젝트가 요구하는 기술의 최신성은 개발자들에게 있어서 프로젝트 참여 의도에 영향을 줄 수 있다. 예를 들어, 소프트

웨어 개발자가 현재의 개발 방법론에 좀 더 편안해 한다면, 프로젝트에서 새로운 방법론을 적용한다고 할 때에 동기가 저하될 수 있다. 반면, 새로움을 추구하거나 자신의 역량을 향상시키려는 사람의 경우는 신기술을 사용해야 하는 프로젝트에 대해 더 고무적이고 자기 발전적인 일로 생각하게 될 것이고, 이는 참여 의도의 증가로 연결될 것으로 생각된다.

본 연구에서는 프로젝트에서 도입하는 기술의 최신성이 개발자의 프로젝트 성격을 인지하는 데에 영향을 준다고 주장한다. 본 연구는 프로젝트의 성격을 조절 초점이론에서 주로 연구되는 일곱가지 차원으로 집중해서 조명하고자 하는데, 이들 차원은 이상/현실 (Ideal/Realistic), 개방/폐쇄 (Open/Closed), 자율/관리 (Autonomous/Controlled), 도전/안정 (Challenge/Stable), 신남/평온 (Excitement/Calm), 이익극대화/손실최소 (Maximize Profit/Minimize Loss), 성취대상/유지대상 (Achieve Object/Maintain Object) 등이다. 본인에게 제안된 프로젝트의 기술이 얼마나 새로운지에 따라, 개발자들은 해당 프로젝트를 심리적/인지적으로 각 차원에 대해서 다르게 느낄 것이다. 따라서 다음과 같은 가설을 제시한다.

가설 1: 프로젝트의 도입기술의 최신성 정도는 개발자가 인지적으로 느끼는 프로젝트 성격에 영향을 준다.

가설 1-1: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 현실적이기 보단 이상적으로 인지할 것이다.

가설 1-2: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 폐쇄적이기 보단 개방적으로 인지할 것이다.

가설 1-3: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 관리적이기 보단 자율적으로 인지할 것이다.

가설 1-4: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 안정적이기 보단 도전적으로

인지할 것이다.

가설 1-5: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 평온하다고 느끼기 보단 신나고 재미있다고 느낄 것이다.

가설 1-6: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 손실최소화라기 보단 이익극대화라고 인지할 것이다.

가설 1-7: 프로젝트의 도입기술이 신규 기술일수록, 개발자들은 프로젝트의 성격을 유지대상이라고 느끼기 보단 성취대상으로 느낄 것이다.

프로젝트에 관한 동기는 프로젝트를 수행하는 개발자의 성향에 따라 달라질 수 있다. Kirton(1994)은 업무를 수행하는 개인을 안정적인 업무 수행을 선호하는 개인과 새로운 업무 수행을 선호하는 개인으로 구분하였고, Hohmann(1996)은 소프트웨어 개발자가 자신이 혁신적 스타일인지, 수용적 스타일인지에 따라 자신에게 맞는 프로젝트를 선택할 때 직무의 만족도가 높다고 하였다(Linberg 1999).

신규기술의 도입은 위험을 감수하는 행동인데, 위험을 감수함으로써 이익을 얻을 가능성을 증가시키는 것으로 향상초점과 관련된다. 반대로 예방초점 성향의 사람들은 위험이라는 것은 손실을 의미하기 때문에, 손실을 피하기 위해 보수적이고 안전한 선택을 하려고 한다(Crowe and Higgins 1997). 이러한 관점에서 프로젝트 도입 기술의 최신성 정도에 대해서 향상초점의 소프트웨어 개발자와 예방초점의 소프트웨어 개발자 간에는 차이가 존재할 것으로 보인다. 도입 기술의 최신성의 주는 이득을 보여줄 경우, 향상초점의 개발자의 참여의도가 높을 것이다. 반대로 이전의 안정적 기술, 즉 이미 검증된 기술이라는 점을 보여줄 경우, 예방초점의 개발자들이 프로젝트의 참여의도가 높아질 것이라고 예상할 수 있다. 따라서 다음의 연구 가설을 제안한다.

가설 2: 프로젝트의 도입기술의 최신성 정도와 소프트웨

어 개발자의 조절초점에 따라 프로젝트 참여의도는 달라질 것이다.

가설 2-1: 프로젝트의 도입기술이 신규 기술일 경우, 향상초점의 개발자들이 예방초점 개발자들보다 참여의도가 높을 것이다.

가설 2-2: 프로젝트의 도입 기술이 기존기술일 경우, 예방초점의 개발자들이 향상초점 개발자들보다 참여의도가 높을 것이다.

3. 연구설계 및 방법

3.1 실험설계 및 실험변수

3.1.1 프로젝트 특성별 시나리오

본 연구는 시나리오 실험법(Scenario Experiment)을 통해서 가설을 검증하고자 한다. 방법론적으로 시나리오 실험법이란 가설적으로 도출된 방법론을 사용하는 것으로, 외생변수를 통제하고 독립변수의 임의적 조작을 통해 독립변수와 종속변수간의 인과관계를 밝힐 수 있는 장점을 가진다. 시나리오 실험법에서는 조작되거나 통제된 변수의 높은 내적 타당성(Internal Validity)을 가진다. 이러한 통제가능성은 모델이나 이론으로부터 도출된 예상들을 연구자가 테스트할 수 있도록 해준다(Judd et al. 1991; Kim and Jang 2014; Calder et

al. 1981).

본 연구에서는 이에 ‘프로젝트 특성별 시나리오’를 작성하였다. 시나리오 작성을 위해 일반적으로 프로젝트 선택에 있어 프로젝트의 어떤 특성이 중요한 영향을 미치는지 알아볼 필요가 있었다. 이에 일선의 소프트웨어 개발자들과 인터뷰를 진행하였다. 인터뷰에 참여한 소프트웨어 개발자들은 대기업의 웹서비스 개발자, 시스템통합 프로젝트를 주로 수행하는 중소기업 개발자, 웹사이트 및 앱을 개발하는 프리랜서 개발자 등 다양한 영역의 개발자를 대상으로 인터뷰를 하였다.

우선 문헌연구를 통해서 IT 프로젝트 성공과 결과에 영향을 미치는 프로젝트 특성을 도출하고, 이러한 특성들을 중심으로 인터뷰를 진행하였고, 실제 현업의 소프트웨어 개발자들이 생각하는 프로젝트 참여와 진행에 있어 중요하게 생각하는 특성들이 무엇인지를 파악하였다. 그리고 실제적인 프로젝트 시나리오를 작성하기 위해 프리랜서 개발자와 프로젝트 발주사를 연결해주는 온라인 아웃소싱 서비스에서 개발 의뢰서들을 분석했다. 인터뷰와 실제 조사를 바탕으로 프로젝트 특성별 시나리오를 작성하였고, 시나리오의 내적 타당성을 높였다. 조작된 시나리오는 현업에 있는 소프트웨어 개발자뿐만 아니라 프로젝트 관리 전문가들을 대상으로 검토를 의뢰하여, 작성한 시나리오의 내용을 점검했다. 각 시나리오의 내용은 <표 1>과 같다.

<표 1> 시나리오별 구성 및 조작내용

시나리오	내용구성	조작화 내용
프로젝트 도입기술 최신성	사용자에게 정보를 제공하는 서비스 개발 프로젝트로, 서비스 개발에 필요한 기술 도입 상황을 제시함.	신규 기술 적용
		기존 기술 적용

3.1.2 자기조절초점

조절초점은 개인의 성격과 같이 한 가지 초점으로 기질적(Chronic)으로 형성 되어 있기도 하나(Camacho et al. 2003; Shah and Higgins 2001), 상황에 따라서

일시적으로 향상초점이나 예방초점으로 조작될 수 있다(Haws et al. 2010; Higgins 1998; Lee et al. 2000). 즉, 한 개인의 조절초점은 이득이나 성취여부가 문제되는 상황에서는 향상초점으로 활성화될 수 있고, 손실과

같이 상실여부가 문제되는 상황에서는 예방초점으로 활성화될 수 있다.

본 연구에서는 소프트웨어 개발자의 기질적인 조절 초점(Regulatory Focus)을 측정하기 위하여 Sassenberg et al.(2012)의 측정문항 중, 항상초점 3문항과 예방초점 3문항을 사용하였고, 리커드 7점 척도를 사용하여 측정하였다.

3.1.3 참여의도

의도(Intention) 또는 행동의도(Behavioral Intention)은 정보시스템분야의 여러 영역, 특히 기술수용과 관련에서 활용되어 왔다(Davis 1989; Koufaris 2002; Taylor and Todd 1995; Venkatesh et al. 2003; Venkatesh 2000; Venkatesh and Davis 2000).

Ajzen(1991)은 계획행동이론(TPB, Theory of Planned Behavior)에서 인간의 행동에서 행동의도가 매우 중요한 요인인데, 의도는 행동에 영향을 미치는 선행 요인이다. 의도는 또한 행동적 태도와 주관적 규범, 지각된 행동 통제 요인에 영향을 받는다. 따라서 본 연구에서는 이러한 이론적 배경에 따라서 프로젝트 참여라는 행동의 선행요인으로 참여의도를 정의하였으며, 기존 문헌의 측정문항을 본 연구에 맞도록 수정하여 사용하였다. 마찬가지로 리커드 7점 척도를 사용하여 측정하였다.

3.2 실험 순서와 참가자

3.2.1 실험 순서

본 연구의 소프트웨어 개발자의 조절초점과 주어지는 프로젝트 특성에 따른 프로젝트 참여의도에 관해서 알아보고자 한다. 따라서 실험 대상은 현업에서 일하고 있는 한국인 소프트웨어 개발자로 참가자를 한정하여 실험을 진행했다. 참가자의 모집은 대기업 및 중소기업의 현업 소프트웨어 개발자에게 직접 온라인 설문지를 제공하고 응답을 받았으며, 추가적으로 소프트

웨어 개발자 커뮤니티(페이스북 페이지)를 통해 온라인 설문 참여자를 모집하였다. 온라인 설문지 유형은 2가지로, 항상초점적 시나리오만 구성된 설문지와 예방초점적 시나리오만 구성된 설문지로, 설문지 유형은 각 참가자에게 임의적으로 제공되었다.

온라인 설문지를 받은 참가자는 우선 연구 취지와 목적에 관한 안내를 받은 뒤, 이어서 시나리오 공통 상황에 대한 내용을 확인하게 된다. 이후 첫번째로, 기질적 조절초점을 확인하는 문항에 대해 응답을 한 뒤, 본격적으로 시나리오를 제시받고, 각 시나리오에 관한 응답을 하였다. 참가자는 시나리오에 대한 조작 점검 문항에 응답을 한뒤에, 프로젝트 참여의도에 관한 응답을 한다. 설문지 마지막에는 인구통계학적 문항에 대해서 응답을 한 뒤에 모든 온라인 설문 및 실험이 종료된다. 온라인 설문 참가자에는 보상으로 모바일 음료 쿠폰을 제공하였다.

3.2.2 실험 참가자의 특성

설문 수집 결과, 총 260건의 설문지 수집되었으며, 응답이 누락되었거나, 불성실 응답을 제외한 207건의 데이터를 분석에 사용하였다. 실험참가자의 대다수는 남자(177명, 88.5%)였으며, 나이는 20대와 30대가 75명과 104명으로 전체 응답자의 다수(86.4%)를 차지하였다. 학력은 대졸이 137명(66.2%)로 가장 많았으며, 전반적으로 참가자의 학력 수준은 높았다. 경력은 대체로 고른 분포를 보였는데, '1~3년 미만'이 46명(22.2%), '5~10년 미만'이 46명(22.2%), '10년 이상'이 54명(26.1%)으로 가장 높게 나타났다.

참가자의 종사분야는 주로 '웹서비스(web service)'가 가장 많았고(72명, 34.8%), 그 다음으로 '시스템통합(system integration) 또는 유지보수(system maintenance)' 분야가 많았다(56명, 27.1%). 주로 사용하는 개발 언어는 'Java'가 압도적으로 많았으며(110명, 53.1%), 그 다음으로 많이 응답한 개발 언어는 'C, C++'였다

(37명, 17.9%). ‘Perl’ 이라고 응답한 참가자는 한명도 없었다. 기타 언어로 응답한 참가자는 20명(9.7%)으로 ‘Python’과 ‘SQL’ 등이 있었다.

4. 분석 및 결과

4.1 실험 참가자의 조절초점

4.1.1 조절초점 문항의 신뢰성과 타당성

조절초점을 묻는 문항의 신뢰성을 확인하고자, 내적 일관성(Internal Consistency) 검정인 Cronbach’s

Alpha 값을 사용하였다. 측정하고자 하는 속성이나 개념이 측정 도구에 의해 제대로 측정되었는지 구성 타당성(Construct Validity)을 검정하기 위해 요인분석을 실시했다.

분석결과는 <표 2>와 같으며, 모든 문항의 요인부하량이 0.7 이상이며, 각 요인의 고유값은 1.0 이상, 두 요인이 설명하는 전체 변동 설명력은 50% 이상으로 조절초점 문항의 타당도는 양호한 것으로 나타났으며, ‘향상초점’과 ‘예방초점’을 측정하는 문항들의 Cronbach’s Alpha 값도 0.6 이상으로 신뢰성에 문제가 없는 것으로 나타났다.

<표 2> 조절초점 문항의 신뢰성과 타당성

구분	문항	요인부하량	고유값	변동 설명력	누적 변동 설명력	Cronbach's Alpha
향상 초점	RF1	0.796	1.984	33.065	33.065	0.739
	RF2	0.801				
	RF3	0.820				
예방 초점	RF4	0.753	1.759	29.314	63.379	0.639
	RF5	0.817				
	RF6	0.704				

4.1.2 향상초점 및 예방초점 집단 구분

향상초점과 예방초점 집단의 구분은 중위수 분리법(Median Split Method)에 의거하여 분리하였다. 향상초점 문항 점수와 예방초점 문항 점수를 뺀 값을 중앙값 기준으로 나누어, 중앙값보다 높은 경우 향상초점 집단으로, 낮은 경우 예방초점 집단으로 나누었다. 향상초점과 예방초점 분리결과, 향상초점 집단은 107명으로 예방초점 집단은 100명으로 구분되었다.

오별로 조작이 되었는지 확인하기 위해, 조작점검 문항의 평균(Eigen Value = 1.871, Cronbach’s Alpha = 0.930)이 향상초점 성향에 적합한 시나리오와 예방초점 성향에 적합한 시나리오 그룹 간에 차이를 ANOVA 검정을 통해 보았다. 향상초점 성향에 적합한 시나리오 그룹의 평균은 5.32이고, 예방초점 성향에 적합한 시나리오 그룹의 평균은 3.01로 두 그룹간에는 통계적으로 유의한 차이를 보였다($F(1, 205) = 125.003, p < 0.001$).

4.2 조절초점에 따른 프로젝트의 도입기술 최신

성과 참여의도 연구 결과

4.2.1 조작점검

프로젝트의 ‘도입기술 최신성’이 의도한 대로 시나리

4.2.2 가설검정

가설 1을 검증하기 위해, 프로젝트 도입기술 최신성 특성에 대해 실험 참가자는 어떠한 성향으로 받아들이고 있는지를 분석해보았다. 실험 참가자가 주어진 프

젝트 특성을 보고, 프로젝트 성향이 어떠한지를 평가하였고, 해당 평가가 프로젝트 특성 간의 차이가 있는지를 ANOVA 분석을 통해 확인하였다. 분석결과는 <표 3>와 같다. 분석결과 일곱 가지의 성향 모두, 유의확률 0.001 이하로 유의한 것으로 나타났다. 따라서 가설 1은 채택되었다.

IT 프로젝트의 참여의도가 프로젝트의 ‘도입기술 최

신성’에 따라 어떻게 달라지를 분석하여 연구 가설 2를 검증하였다. 참여의도 3문항을 평균하여 도출된 각 조건의 평균과 표준편차는 <표 4>와 같다. 가설검정을 위해서는 일변량 일반선형모형을 분석하였다. 결과 <표 5>와 같이 나왔으며, 프로젝트의 도입기술 최신성과 조절초점간의 상호작용은 유의하게 나타났다($F(1, 205)=19.303, p < 0.001$).

<표 3> 프로젝트 도입기술 최신성에 따른 인지된 프로젝트 성격

구분	F	p
이상/현실(Ideal/Realistic)	80.790	.000***
개방/폐쇄(Open/Closed)	42.433	.000***
자율/관리(Autonomous/Controlled)	67.610	.000***
도전/안정(Challenge/Stable)	125.496	.000***
흥분/평온(Excitement/Calm)	103.050	.000***
이익극대화/손실최소(Maximize Profit/Minimize Loss)	54.601	.000***
성취대상/유지대상(Achieve Object/Maintain Object)	90.388	.000***

* $p < .05$, ** $p < .01$, *** $p < .001$

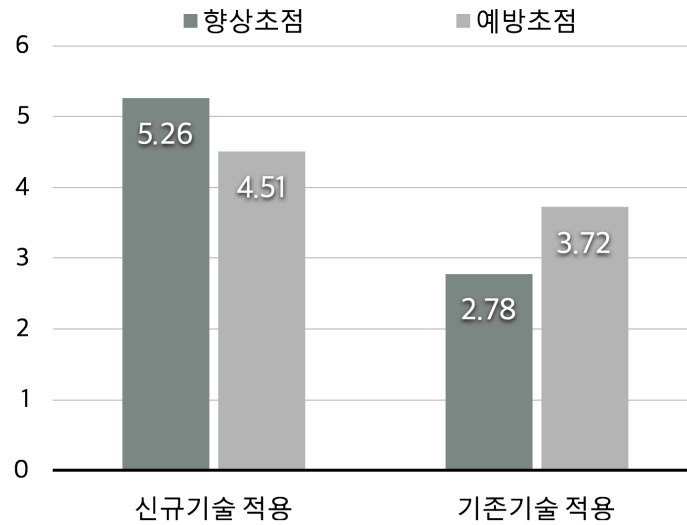
<표 4> 도입기술 최신성과 참여의도 기술통계

구분	프로젝트 도입기술 최신성	
	신규기술 적용	기존기술 적용
향상초점	5.26 (1.35) n=59	2.78 (1.53) n=48
예방초점	4.51 (1.33) n=41	3.72 (1.32) n=59

<표 5> 도입기술 최신성과 참여의도 일변량 일반선형모형 결과

구분	참여의도	
	F	p
(A) 프로젝트 특성(도입기술 최신성)	73.352	.000***
(B) 조절초점	.200	.655
(A) × (B)	19.303	.000***

* $p < .05$, ** $p < .01$, *** $p < .001$



<그림 1> 도입기술 최신성에 대한 참여의도 변화

<그림 1>에서 확인할 수 있듯이, 프로젝트가 신규기술을 적용할 경우, 향상초점 개발자의 참여의도(평균=5.26)가 예방초점 개발자의 참여의도(평균=4.51)보다 높게 나타났다($F(1, 98)=0.750, p < 0.01$). 프로젝트가 기존기술을 적용할 경우, 향상초점 개발자의 참여의도(평균=2.78)보다 예방초점 개발자의 참여의도(평균=3.72)가 높게 나타났다($F(1, 105)=0.644, p < 0.05$). 따라서 연구 가설 1을 모두 지지한 결과가 나왔으며, 신규기술을 적용하는 프로젝트의 경우에는 향상초점 개발자의 참여의도가 높고, 기존기술을 적용하는 프로젝트의 경우에는 예방초점 개발자의 참여의도가 높다는 것을 알 수 있다.

4.2.3. 논의

신규기술을 적용하는 프로젝트에 대해서 향상초점 성향의 개발자들이 참여의도가 높게 나타났으며, 기존기술을 적용하는 프로젝트에 대해서는 예방초점 성향의 개발자들이 상대적으로 높은 참여의도를 보였다. 그러나 전반적으로 향상초점 및 예방초점 개발자 모두 신

규기술을 적용하는 프로젝트에 대한 참여의도가 기존기술 적용 프로젝트보다 상대적으로 높은 참여의도 값을 보였다.

이는 기존 연구들에서 이미 밝혀진 바와 같이, 전반적으로 소프트웨어 개발자들은 새로운 솔루션이나 디자인, 기술적 도전, 새로운 기술과 방법론을 사용하는 것을 선호한다는 사실을 다시 한번 확인할 수 있었다. 프로젝트 성향에 대한 분석대로, 개발자들은 신규기술의 사용에 대해서 감정적으로 '흥분'과 '도전적' 상태, 그리고 '개방적' 상태로 동기가 유발된다고 볼 수 있다.

그러나 향상초점 성향의 개발자들의 경우에는 기존기술을 사용하는 프로젝트가 주어질 경우, 동기유발이 되지 않는 것으로 보인다. 실험결과를 보면, 모든 조건의 참여의도 값 중에서도 기존기술을 사용하는 프로젝트에서의 향상초점 성향 개발자들의 참여의도가 가장 낮게 나타났다. 이는 프로젝트 관리 관점에서 기존기술을 사용해야 하는 프로젝트에서 향상초점 성향 개발자들의 동기를 어떻게 올릴 수 있을지 실무적 고민을 던져준다.

5. 결론

본 연구의 목적은 조절초점이론을 통해 예측한 바와 같이 소프트웨어 개발자에게도 주어지는 프로젝트의 조건에 따라 소프트웨어 개발자의 성향에 따른 차이가 나타나는지 알아보는 것이었다. 우선 향상초점 성향의 개발자들은 참여할 프로젝트가 '신규기술'을 적용할 경우, 예방초점의 개발자보다 높은 참여의도를 보였다. 이는 향상초점 성향의 사람들은 목표에 대한 성취, 향상, 증진, 열망과 같은 진취적 성향을 가지고 있고, 새로운 상황에 대한 도전 의식이 있고, 향상초점적 성향의 수단이나 정보를 접할 때에 적합성이 일어나, 이에 대해 높은 가치를 두어 평가한다는 조절초점 이론의 설명과 일치한다(Aaker and Lee 2001; Higgins 1997, 2000).

반대로 예방초점 성향의 개발자들은 참여할 프로젝트가 '기존기술'을 사용할 경우, 향상초점 개발자보다 높은 참여의도를 보였다. 이는 안전, 책임, 보호, 방어, 회피적인 성향, 변화나 새로운 상황에 대한 소극적이며, 실패에 민감하여 최소한의 목표를 추구하는 성향을 가진 예방초점 소프트웨어 개발자들에게는 안정적인 '기존기술'을 사용하는 프로젝트에 대한 적합성(fit)이 일어난 것으로 보이며, 이는 조절초점이론에서의 예방초점 성향의 특징과 일치하는 점이다(Brendl and Higgins 1996; Crowe and Higgins 1997; Higgins 1997; Lee et al. 2000).

그러나 '신규기술'에 대한 참여의도가 '기존기술'을 사용하는 경우보다, 향상초점 및 예방초점 개발자 모두 참여의도 평균값이 높게 나타났다. 이는 기존 연구에서 말하는 소프트웨어 개발자들의 독립적이고 자율을 추구하는 일반적인 향상적 특징이 반영된 것으로 생각할 수 있다(Beecham et al. 2008; Couger and Zawacki 1980).

본 연구는 우선 Higgins의 조절초점이론을 확장하는데 기여를 하였다. 조절초점이론의 연구들이 주로 마

케팅과 소비자행동, 서비스 분야에서 가장 활발히 응용되고 있는데, 본 연구는 조절초점이론을 소프트웨어 공학 분야로 확장했다는 데에 의의가 있다. 지금까지는 소프트웨어 개발자들에 대한 동기와 특성 연구에 있어서 주로 MBTI와 고전적 동기이론을 중심으로만 연구가 되어왔다(Capretz 2003; Cruz et al. 2015; Feldt et al. 2010; Hall et al. 2009; Smith 1989; Turley and Bieman 1995). 본 연구에서는 Higgins의 조절초점이론을 바탕으로 기존 연구들이 사용하지 않았던 새로운 이론적 접근으로 소프트웨어 개발자의 성향을 분석하고, IT 프로젝트 참여의도에 미치는 영향에 대해서 분석하였다.

나아가 본 연구는 '행동 소프트웨어 공학(behavioral software engineering)'에 속한 연구로써 이론적 기여를 할 수 있다고 본다. 소프트웨어 공학 분야가 이론의 사용과 실증적 작업이 부족하고, 다양한 분야의 이론을 적용한 연구가 부족하다는 지적이 있어왔다(Hannay et al. 2007; Kitchenham et al. 2004; Lenberg et al. 2015). 이에 본 연구는 타 분야야의 이론을 사용하여 소프트웨어 공학 분야의 확장에 이론적 기여를 하였다.

실무적으로 본 연구는 IT 관련 기업에서 소프트웨어 개발자를 이해하고, 프로젝트 관리에 도움이 될 수 있는 결과들을 제시하였다. 기존의 연구에서와 마찬가지로, 소프트웨어 개발자의 동기가 프로젝트 수행과 관련된 행위에 중요한 역할을 한다는 사실을 본 연구에서도 다시 한번 확인할 수 있었다. 이러한 개발자의 동기를 고려하지 않은 상태에서 프로젝트를 진행할 경우, 프로젝트의 성과를 기대하기 어려울 수 있다.

구체적으로 중요한 발견은 다음과 같다. 향상초점과 예방초점 개발자 간에 프로젝트 특성에 의한 참여의도 차이는 존재하나, 전반적으로 개발자들은 성취 지향적이며, 성장과 향상의 욕구를 가지고 있다고 볼 수 있다. 본 연구의 실험 결과에 의하면 향상초점 및 예방초점

개발자 모두, '신규기술'을 사용하는 프로젝트의 참여의도가 높게 나타났다. 따라서, 소프트웨어 개발자의 참여의도를 높이고, 동기부여를 통해 프로젝트의 성과를 높이기 위해서는 프로젝트의 성격이나 목표를 항상적으로 제시할 필요가 있다.

다음으로 항상초점 성향과 예방초점 성향의 소프트웨어 개발자의 특성에 맞는 관리적 방법을 달리 추구할 필요가 있다. 예방초점 성향의 소프트웨어 개발자는 프로젝트 특성에 따른 참여의도가 크게 차이가 나지 않았다. 그러나 예방초점 성향의 개발자들도 자신의 성향과 일치하는 프로젝트(본 연구에서는 '기존기술'을 사용하는 프로젝트)에 대해서 참여의도가 크므로, 그러한 성향을 고려한 프로젝트의 부여, 업무배치를 고려해볼 수 있다.

이에 반해 항상초점 개발자들은 성취지향적이고, 결과에 대한 욕망으로 우수한 결과들을 산출해 낼 수 있는 사람들이지만, 실험의 결과에서 보여주었다 시피, 프로젝트 특성의 변화에 의해 참여의도의 동기가 크게 낮아지는 것을 볼 수 있었다. 따라서 항상초점 개발자들의 생산성과 참여의도를 높이기 위해서 지속적인 동기부여뿐만 아니라 좌절할 수 있는 프로젝트 요건을 만들지 않도록 해야할 것이다.

본 연구는 다음과 같은 한계점을 가진다. 우선 시나리오 실험법이라는 방법론적 한계가 존재한다. 시나리오 실험법은 실험에서의 사람들의 행위와 실제 현실에서의 행위와 연계성이 없다는 비판을 받고 있다(Falk and Heckman 2009). 또한 시나리오 실험법에서의 인위적인 조건이 비현실적인 데이터를 만들어 낸다는 비판도 있다(Bardsley 2005). 이는 실험연구방법론에서 실험자의 행위가 원하는 방향으로 실험 대상자가 순응하기 때문에 생기는 문제이다(Levitt and List 2007; Orne 1962).

두 번째로, IT 개발 프로젝트가 가지는 다양한 요인들을 살펴볼지 못한 부분이 있다. 본 연구에서는 '도입

기술의 최신성'만을 프로젝트의 특성으로 설정하고 실험을 하였다. 그러나 소프트웨어 개발의 성공에 영향을 주는 요인은 이 외에도 조직적 차원, 외부환경적 조건, 소프트웨어 및 시스템 사용자 등에서도 영향을 받는다(McLeod and MacDonell 2011). 그리고 개발자의 경우에도 프로젝트 특성 이외에도 다양한 내재적, 외재적 동기요인을 통해서 영향을 받는다(Sharp et al. 2009). 따라서 이러한 한계점을 보완하여 향후 연구에서는 IT 개발 프로젝트를 둘러싼 다른 요인과 개발자 간의 관계에 대한 추가적인 연구가 필요하다.

세 번째로, 제한적인 실험 대상자만을 설정하여 연구를 진행한 한계점이 존재한다. 소프트웨어 산업이 다양해지면, 다양한 분야의 다양한 직군의 IT 인력이 생겨나고 있다. 각 분야와 업무 직군에 따라 개발자의 성향이 다를 수 있다. 예를 들어 시스템 유지보수 업무 분야에서 일하는 개발자는 게임을 개발하는 개발자와 성향이 다를 수 있다. 전자는 업무의 변화 가능성이 낮고, 안정적인 운영을 목적으로 하기 때문에, 예방초점적 성향을 가질 수 있다. 또는 업무적 특성에 따라서도 달라질 수 있다. 사용자의 인터페이스 등, 눈에 보이는 요소를 다루는 프론트엔드(Front-end) 개발자는 항상초점 성향이 강하다고 예상할 수 있다. 반면, 시스템 하부의 안정적이고, 자원의 최적화를 다루는 백엔드(Back-end) 개발자의 경우, 예방초점 성향이 강하다고 예상할 수 있다. 따라서 향후 연구에서는 소프트웨어 개발자의 직군과 직무에 따른 세부적인 연구가 필요하다.

참고문헌

1. Aaker, J. L. and Lee, A. Y. 2001, ““I” seek pleasures and “we” avoid pains: The role of self-regulatory goals in information processing and persuasion”, *Journal of Consumer Research*(28:1), pp. 33-49.
2. Ajzen, I. 1991, “The theory of planned behavior”, *Organizational Behavior and Human Decision Processes*(50:2), pp. 179-211.
3. Bardsley, N. 2005, “Experimental economics and the artificiality of alteration”, *Journal of Economic Methodology*(12:2), pp. 239-51.
4. Beecham, S., Baddoo, N., Hall, T., Robinson, H. and Sharp, H. 2008, “Motivation in Software Engineering: A systematic literature review”, *Information and Software Technology*(50:9), pp. 860-78.
5. Boehm, B. W. 1981, *Software engineering economics*, Prentice-hall Englewood Cliffs (NJ).
6. Boehm, B. W. 1988, “Understanding and controlling software costs”, *Journal of Parametrics*(8:1), pp. 32-68.
7. Bradley, J. H. and Hebert, F. J. 1997, “The effect of personality type on team performance”, *Journal of Management Development*(16:5), pp. 337-53.
8. Brendl, C. M. and Higgins, E. T. 1996, “Principles of judging valence: What makes events positive or negative?”, *Advances in experimental social psychology*(28), pp. 95-160.
9. Butler, T. and Fitzgerald, B. 2001, “The Relationship between User Participation and the Management of Change Surrounding the Development of Information Systems: A European Perspective”, *Journal of End User Computing*(13:1), pp. 12-25.
10. Calder, B. J., Phillips, L. W. and Tybout, A. M. 1981, “Designing research for application”, *Journal of Consumer Research*(8:2), pp. 197-207.
11. Camacho, C. J., Higgins, E. T. and Luger, L. 2003, “Moral value transfer from regulatory fit: what feels right is right and what feels wrong is wrong”, *Journal of Personality and Social Psychology*(84:3), pp. 498.
12. Capretz, L. F. 2003, “Personality types in software engineering”, *International Journal of Human-Computer Studies*(58:2), pp. 207-14.
13. Cohen, S. G. and Ledford, G. E. 1994, “The effectiveness of self-managing teams: A quasi-experiment”, *Human relations*(47:1), pp. 13-43.
14. Constantine, L. L. 1995, *Constantine on peopleware*, Prentice Hall.
15. Couger, J. D. and Zawacki, R. A. 1980, *Motivating and managing computer personnel*, Wiley.
16. Crowe, E. and Higgins, E. T. 1997, “Regulatory focus and strategic inclinations: Promotion and prevention in decision-making”, *Organizational Behavior and Human Decision Processes*(69:2), pp. 117-32.

17. Cruz, S., da Silva, F. Q. B. and Capretz, L. F. 2015, "Forty years of research on personality in software engineering: A mapping study", *Computers in Human Behavior*(46), pp. 94-113.
18. Curtis, B., Krasner, H. and Iscoe, N. 1988, "A field study of the software design process for large systems", *Communications of the ACM*(31:11), pp. 1268-87.
19. Da Silva, F. Q. B. and França, A. C. C. 2012, "Towards understanding the underlying structure of motivational factors for software engineers to guide the definition of motivational programs", *Journal of Systems and Software*(85:2), pp. 216-26.
20. Davis, F. D. 1989, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*(13:3), pp. 319-40.
21. Falk, A. and Heckman, J. J. 2009, "Lab experiments are a major source of knowledge in the social sciences", *Science*(326:5952), pp. 535-38.
22. Faraj, S. and Sproull, L. 2000, "Coordinating expertise in software development teams", *Management Science*(46:12), pp. 1554-68.
23. Feldt, R., Angelis, L., Torkar, R. and Samuelsson, M. 2010, "Links between the personalities, views and attitudes of software engineers", *Information and Software Technology*(52:6), pp. 611-24.
24. Fitzgerald, B. 1998, "An empirically-grounded framework for the information systems development process", in *Proceedings of the international conference on Information systems*, Association for Information Systems, pp. 103-14.
25. Fitzgerald, B. and Fitzgerald, G. 1999, "Categories and Contexts of Information Systems Development: Making Sense of the Mess", in *ECIS*, Citeseer, pp. 194-211.
26. Guinan, P. J., Coopridge, J. G. and Sawyer, S. 1997, "The effective use of automated application development tools", *IBM Systems Journal*(36:1), pp. 124-39.
27. Hall, T., Baddoo, N., Beecham, S., Robinson, H. and Sharp, H. 2009, "A systematic review of theory use in studies investigating the motivations of software engineers", *ACM Transactions on Software Engineering and Methodology (TOSEM)*(18:3), pp. 10.
28. Hannay, J. E., Sjoberg, D. I. K. and Dyba, T. 2007, "A systematic review of theory use in software engineering experiments", *IEEE Transactions on Software Engineering*(33:2), pp. 87-107.
29. Haws, K. L., Dholakia, U. M. and Bearden, W. O. 2010, "An assessment of chronic regulatory focus measures", *Journal of Marketing Research*(47:5), pp. 967-82.
30. Higgins, E. T. 1997, "Beyond pleasure and pain", *American Psychologist*(52:12), pp. 1280.
31. Higgins, E. T. 1998, "Promotion and prevention: Regulatory focus as a motivational principle", *Advances in experimental social psychology*(30), pp. 1-46.

32. Higgins, E. T. 2000, "Making a good decision: value from fit", *American Psychologist*(55:11), pp. 1217.
33. Higgins, E. T., Shah, J. and Friedman, R. 1997, "Emotional responses to goal attainment: strength of regulatory focus as moderator", *Journal of Personality and Social Psychology*(72:3), pp. 515.
34. Hohmann, L. 1996, *Journey of the software professional: a sociology of software development*, Prentice-Hall.
35. Janz, B. D., Colquitt, J. A. and Noe, R. A. 1997, "Knowledge worker team effectiveness: The role of autonomy, interdependence, team development, and contextual support variables", *Personnel Psychology*(50:4), pp. 877-904.
36. Jones, M. C. and Harrison, A. W. 1996, "IS project team performance: an empirical assessment", *Information & Management*(31:2), pp. 57-65.
37. Judd, C. M., Smith, E. R. and Kidder, L. H. 1991, *Research methods in social relations*, Holt, Rinehart, and Winston.
38. Kautz, K. 2004, "The enactment of methodology: the case of developing a multimedia information system", *ICIS 2004 Proceedings*, pp. 55.
39. Kim, J.-H. and Jang, S. S. 2014, "A scenario-based experiment and a field study: A comparative examination for service failure and recovery", *International Journal of Hospitality Management*(41), pp. 125-32.
40. Kirton, M. J. 1994, "Adapters and Innovators: Styles of Creativity and Problem Solving", in, Routledge, New York.
41. Kitchenham, B. A., Dyba, T. and Jorgensen, M. 2004, "Evidence-based software engineering", in *Proceedings of the 26th international conference on software engineering*, IEEE Computer Society, pp. 273-81.
42. Koufaris, M. 2002, "Applying the technology acceptance model and flow theory to online consumer behavior", *Information Systems Research*(13:2), pp. 205-23.
43. Lee, A. Y., Aaker, J. L. and Gardner, W. L. 2000, "The pleasures and pains of distinct self-construals: the role of interdependence in regulatory focus", *Journal of Personality and Social Psychology*(78:6), pp. 1122.
44. Lenberg, P., Feldt, R. and Wallgren, L. G. 2015, "Behavioral software engineering: A definition and systematic literature review", *Journal of Systems and Software*(107), pp. 15-37.
45. Levitt, S. D. and List, J. A. 2007, "Viewpoint: On the generalizability of lab behaviour to the field", *Canadian Journal of Economics/Revue canadienne d'économique*(40:2), pp. 347-70.
46. Liberman, N., Idson, L. C., Camacho, C. J. and Higgins, E. T. 1999, "Promotion and prevention choices between stability and change", *Journal of Personality and Social Psychology*(77:6), pp. 1135.

47. Linberg, K. R. 1999, "Software developer perceptions about software project failure: a case study", *Journal of Systems and Software*(49:2), pp. 177-92.
48. Lyytinen, K. and Hirschheim, R. 1987, "Information systems failures: a survey and classification of the empirical literature", *Oxford Surveys in Information Technology*(4:1), pp. 257-309.
49. McConnell, S. 1998, "Problem programmers", *IEEE Software*(15:2), pp. 126-28.
50. McLeod, L. and MacDonell, S. G. 2011, "Factors that affect software systems development project outcomes: A survey of research", *ACM Computing Surveys (CSUR)* (43:4), pp. 24.
51. Myers, I. B., McCaulley, M. H., Quenk, N. L. and Hammer, A. L. 1998, *MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator*, Consulting Psychologists Press Palo Alto, CA.
52. Nandhakumar, J. and Avison, D. E. 1999, "The fiction of methodological development: a field study of information systems development", *Information Technology & People*(12:2), pp. 176-91.
53. Orne, M. T. 1962, "On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications", *American Psychologist*(17:11), pp. 776.
54. Oz, E. and Sosik, J. J. 2000, "Why information systems projects are abandoned: a leadership and communication theory and exploratory study", *Journal of Computer Information Systems*(41:1), pp. 66-78.
55. Poulymenakou, A. and Holmes, A. 1996, "A contingency framework for the investigation of information systems failure", *European Journal of Information Systems*(5:1), pp. 34-46.
56. Procaccino, J. D., Verner, J. M., Overmyer, S. P. and Darter, M. E. 2002, "Case study: factors for early prediction of software development success", *Information and Software Technology*(44:1), pp. 53-62.
57. Procaccino, J. D., Verner, J. M., Shelfer, K. M. and Gefen, D. 2005, "What do software practitioners really think about project success: an exploratory study", *Journal of Systems and Software*(78:2), pp. 194-203.
58. Rasch, R. H. and Tosi, H. L. 1992, "Factors affecting software developers' performance: an integrated approach", *MIS Quarterly*(16:3), pp. 395-413.
59. Roese, N. J., Hur, T. and Pennington, G. L. 1999, "Counterfactual thinking and regulatory focus: Implications for action versus inaction and sufficiency versus necessity", *Journal of Personality and Social Psychology*(77:6), pp. 1109.
60. Sassenberg, K., Ellemers, N. and Scheepers, D. 2012, "The attraction of social power: The influence of construing power as opportunity versus responsibility", *Journal of Experimental Social Psychology*(48:2), pp. 550-55.

61. Schwalbe, K. 2015, *Information technology project management*, 7 ed., Cengage Learning.
62. Scott, J. E. and Vessey, I. 2002, "Managing risks in enterprise systems implementations", *Communications of the ACM*(45:4), pp. 74-81.
63. Shah, J. and Higgins, E. T. 1997, "Expectancy× value effects: Regulatory focus as determinant of magnitude and direction", *Journal of Personality and Social Psychology*(73:3), pp. 447.
64. Shah, J. and Higgins, E. T. 2001, "Regulatory concerns and appraisal efficiency: the general impact of promotion and prevention", *Journal of Personality and Social Psychology*(80:5), pp. 693.
65. Sharp, H., Baddoo, N., Beecham, S., Hall, T. and Robinson, H. 2009, "Models of motivation in software engineering", *Information and Software Technology*(51:1), pp. 219-33.
66. Smith, D. C. 1989, "The personality of the systems analyst: an investigation", *ACM SIGCPR Computer Personnel*(12:2), pp. 12-14.
67. Spector, P. E. 1986, "Perceived control by employees: A meta-analysis of studies concerning autonomy and participation at work", *Human Relations*(39:11), pp. 1005-16.
68. Taylor, S. and Todd, P. 1995, "Assessing IT usage: The role of prior experience", *MIS Quarterly*(19:4), pp. 561-70.
69. Turley, R. T. and Bieman, J. M. 1995, "Competencies of exceptional and nonexceptional software engineers", *Journal of Systems and Software*(28:1), pp. 19-38.
70. Venkatesh, V. 2000, "Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model", *Information Systems Research*(11:4), pp. 342-65.
71. Venkatesh, V. and Davis, F. D. 2000, "A theoretical extension of the technology acceptance model: Four longitudinal field studies", *Management Science*(46:2), pp. 186-204.
72. Venkatesh, V., Morris, M. G., Davis, G. B. and Davis, F. D. 2003, "User acceptance of information technology: Toward a unified view", *MIS Quarterly*(27:3), pp. 425-78.

저 자 소 개



박요한 (Yohan Park)

국민대학교 비즈니스IT전문대학원에서 석사학위를 취득하였다. 현재 NHN comic Japan에서 서비스기획 업무를 맡고 있다. 사용자 경험 기반의 서비스 기획과 Growth Hacking 기법, 데이터 분석과 시각화에 관심을 가지고 있다.



박도형 (Do-Hyung Park)

KAIST 경영대학원에서 MIS 전공으로 석사/ 박사학위를 취득하였다. 현재 국민대학교 경영대학 경영정보학부 조교수로 재직 중이며, 한국과학기술정보연구원(KISTI)에서 유망아이템 발굴, 기술가치 평가 및 로드맵 수립, 빅데이터 분석 등을 수행하였고, LG전자에서 통계, 시선/뇌파 분석, 데이터 마이닝 등 활용한 연구 및 소비자 평가 모형 개발을 담당했었고, 스마트폰, 스마트TV, 스마트Car 등에 대한 Technology, Business, Market Insight 기반 컨셉 도출 프로젝트를 다수 수행하였다. 현재 주요 관심분야는 SNS나 온라인 구전 등의 사용자 행동 이론(User Behavior), 사용자 경험 디자인 프로세스 및 혁신 제품 발굴(User eXperience), 빅데이터 기반 사용자 분석(User Analytics) 등이다.