

# 소프트웨어 개발에서 심리학의 중요성

## (The Importance of Psychology aspect of Software Development)

소정연<sup>†</sup>      최창진<sup>‡</sup>      한승범<sup>†</sup>      민상윤<sup>‡</sup>  
 (Jungyeon Soh) (Changjin Choi) (Seungbum Han) (Sangyoon Min)

**요 약** 소프트웨어의 규모가 커지고 지식 사회로 전환됨에 따라, ‘사람’에 대한 중요성이 대두되고 있다. 소프트웨어 공학에서도 People, Process, Technology 모두 중요하다고 언급이 되지만, 실무에서는 People 측면에 대한 관심이 부족하다. 다른 학문 분야에서는 사람의 심리를 이해하기 위해 여러 가지 방법들이 제시되었다. 본 논문에서는 다른 학문들에서는 어떻게 사람에 대한 심리를 연구하고 있는지에 대해서 네 가지 관점으로 정리하였다. 첫째로, 사람과 사람에 대한 심리학 적용에 대한 멘토링을 다룬다. 둘째로, 팀에 대한 것으로 MBTI를 이용한 조직구성에 대해서 논의한다. 셋째로, 대중에 대한 심리학 적용 사례인 프로파간다를 살펴보고 넷째로, 새로운 소셜 미디어에 관한 내용을 다룬다. 각 방법에서 소프트웨어 개발 분야로 이런 기법을 재해석하여 활용하기 위해 어떤 부분을 조심해야 하는지 살펴보고자 한다.

**키워드** 심리학, 멘토링, MBTI, 프로파간다, 소셜 미디어

**Abstract** Since the industrial society has emerged into the knowledge society and the development of software has enlarged its scale, the importance of the 'people' has surfaced at its peak. Even though the People, the Process and the Technology are mentioned as the core in the field of software engineering, the pragmatic aspect of it tends to lack adequate interest in the People. In other disciplines, several methods have been proposed in order to understand the psychology of the people. In this paper, we have organized the psychological researches that other fields have studied into four different perspectives. First, it covers how to apply a person-to-person psychological method referred to as mentoring. Second, we discuss about a team that utilizes the MBTI system in organizing their structure. Third, we look at a case of the propaganda; the psychological application for the masses. Finally, we cover on the new social media. We investigate on the possible considerations in applying the psychological techniques in the field of software development to utilize the advantages from each methods.

**Key words** : Psychology, Mentoring, MBTI, Propaganda, Social Media

## 1. 서론

우리는 지금까지 소프트웨어 개발 프로젝트를 진행해오면서 사람들 간의 관계 때문에 프로젝트의 성공 실패가 나뉘는 경우를 종종 경험해 왔다. 소프트웨어 공학에서는 프로젝트의 성공 요인을 ‘People’, ‘Process’, ‘Technology’ 3가지 측면에서

<sup>†</sup> 비 회 원 : KAIST 소프트웨어대학원 / (주)삼성전자  
jungyeon.soh@gmail.com

<sup>‡</sup> 비 회 원 : KAIST 소프트웨어 대학원  
mmchang76@gmail.com

<sup>†</sup> 비 회 원 : KAIST 소프트웨어대학원 / 고영테크놀러지 연구소  
vance@kasit.ac.kr

<sup>#</sup> 회 원 : KAIST 소프트웨어대학원 / ㈜솔루션링크  
sang@sol-link.com

바라봐야 한다. 그러나 실무에서는 ‘Technology’와 ‘Process’는 어떻게 개선시킬 수 있을까에 대해서 고민을 해보지만 유독 ‘People’의 측면은 어떻게 개선시킬지에 대해서 고민하기 보다는 단지 사람의 문제라고 생각하는 경우가 많다.

실례로 IEEE 학회에서 발표된 “Why Software Fails”[1]라는 논문을 보면 12개의 실패 원인은 복합적으로 나타난다. 그 중에서 다음과 같은 6가지 원인은 사람에 대해서 이해할 수 있어야 좀 더 원활하게 해결될 수 있다.

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Poor reporting of the project’s status
- Poor communication among customers, developers, and users
- Sloppy development practices
- Poor project management

다른 학문에서는 이러한 문제들을 개선하기 위해서 멘토링과 같은 인재 육성 인간 관계 활동, MBTI 분석과 같은 성격 유형별 관리, 선전 기법 등과 같은 여러 가지 심리 기법이 활발하게 연구되고 있는데 본 논문에서는 이런 기법들이 다른 학문 분야에서는 어떻게 사용되고 있고 소프트웨어 업계에서 어떻게 사용하면 성공할 수가 있는 것일까에 대해서 검토 해 보았다.

본 연구의 구성은 1장 서론에 이어, 2장에서는 소프트웨어 공학에 고려되지 못했던 것이 어떤 것이 있었는지를 살펴보고, 3장에서는 다른 학문들에서는 어떻게 사람에 대한 심리에 대해 연구되고 있는지에 대해서 정리한 후, 4장에서 결론을 제시한다.

## 2. 소프트웨어 공학의 ‘Missing Points’

소프트웨어 공학은 1968년 NATO 소프트웨어 공학 학회에서 소프트웨어 위기(Software crisis)[2]를 심각하게 고려하기 위해서 출현되었다. 소프트웨어 위기의 원인으로 언급된 내용을 보면 다음과 같다.

- 소프트웨어 규모의 대형화
- 복잡화에 따른 개발 비용 증대
- 유지보수의 어려움과 개발적체 현상
- 프로젝트 개발 및 소요예산 예측의 어려움

중요 문제의 맥락은 소프트웨어 규모가 증가되면서 발생한 문제이다. 그 당시 문제를 해결하기 위해 프로세스적인 방법이 도입이 되었고 사람들의 관심을 받으며 발전하게 되었다. 자동화가 되어 있지 않은 프로젝트는 그 규모가 커지면 투입되는 인력도 늘어 날 수밖에 없다. 이런 상황에 맞춰서 인력관리에 대한 관심은 프로세스 분야에 대한 관심에 비해 현저하게 떨어지고 있다.

최근 많이 사용되고 있는 CMMI(Capability Maturity Model Integration)와 P-CMM(People Capability Maturity Model)을 비교해 보아도 알 수가 있다. 두 모델 모두 Carnegie-Mellon 대학의 SEI(Software Engineering Institute) 연구소에서 만들어진 모델로 CMMI는 조직의 개발 프로세스를, P-CMM은 조직의 인적 역량을 평가를 강화하기 위한 모델이다. 출발 시점부터 P-CMM이 CMMI에 비해 5년 늦게 만들어졌으며, 2009년부터 2012년까지 인증 받은 횟수는 P-CMM Ver2 19건, CMMI-Dev v1.2 2700건으로 기업에서도 P-CMM에 대한 관심이 떨어진다는 것을 알 수 있다.

SEI에서 제공되는 P-CMM V2 모델 가이드 문서가 CMMI-Dev v1.2에 비해 157페이지가 많음에도 불구하고 Wikipedia에서 다루고 있는 비중이 낮다 (CMMI가 22 Kbytes, P-CMM의 5 Kbytes). 또한 편집에 참여한 인원도 CMMI는 100명 이상의 유저가 참여했으나 P-CMM은 45명의 유저만 참여하여 일반 사용자들의 관심 역시 부족하다는 것을 알 수 있다.

1957년에 ACM 저널에 발표된 "Psychological Tests and Selection of Computer Programmers"[3] 논문을 보면 소프트웨어 분야에서도 심리학에 대해서 연구가 소프트웨어 공학 이전부터 나왔었고 1971년 소프트웨어 공학 최초의 베스트셀러인 "The Psychology of Computer Programming"[4]이라는 책이 나오면서 프로그래밍 심리학이라는 분야가 만들어졌다. 이 책에서는 '프로그래밍도 사람이 하는 것'이라는 당연하지만 현실에서는 제대로 인정받지 못하는 이 문제 인식을 바탕으로 쓰여진 책이다. 하지만 1998년 25주년 기념판이 출시되었을 때도 원문은 전혀 수정되지 않고 매장마다 98년 시점에서의 짧은 해결을 곁들인 정도로 책이 출판되었다는 것은 얼마나 프로그래밍 심리학이 현실에서 적용되지 않고 있는가를 반영하고 있다.

1975년 "The Mythical Man-Month"[5]라고 하는 프로그래밍 심리학 책이 출판되었다. 이 책은 저자인 Fred Brooks가 IBM에서 OS/360 개발 Project Manager를 하면서 경험했던 자신의 느낌을 에세이 형식으로 쓰여진 책이다. 이 책에서 Brooks의 법칙 "adding manpower to a late software project makes it later"이 언급 되었다. 이 말은 소프트웨어 프로젝트에서 단순히 납기를 지키기 위해서 사람을 투입한다고 해서 문제가 해결되지 않는다는 것이 말한다. 이 책을 출판되면서 개발자들 사이에서는

경험을 통해서 알고 있었던 '월간투입인력 미신'이 대중들에게 알려지게 되었다.

"The Mythical Man-Month"책의 Brooks의 법칙에서는 업무 현장에서는 새로 투입된 인원이 업무에 익숙해 질 때까지 필요한 학습 곡선이 필요하고 인력이 증가할수록 커뮤니케이션의 복잡성이 증가하기 때문이라는 내용을 담고 있다. 또한 "The causes of project failure"에서 사람의 Personnel은 프로젝트를 성공시킬 수도 있고 실패시킬 수도 있는 요인이라고 언급되고 있다. 37년이 지난 현재 실무에서도 사람을 많이 투입해서 빨리 끝나기만을 기대하는 일이 아직도 발생하고 있다는 것을 소프트웨어 관련 업무에 있는 사람들이라면 누구나 경험해 봤을 것이다.

최근에 많이 주목을 받고 있는 애자일 중의 스크럼[6]은 존중과 헌신, 개방과 집중 그리고 믿음과 용기이며 소외된 개인이 아니라 자기 조직화된 구조에서 각각의 경력과 경험이 다른 사람들이 모여 서로 존중하며 진행하는 것이 스크럼의 진정한 가치로 생각하고 있는데 이런 사상이 만들어지게 된 이유는 기존에 사람을 고려하지 못한 환경이 만들어낸 결과라고 생각된다.

### 3. 심리 관련 방법들과 소프트웨어 개발에 적용시 고려해야 할 사항

소프트웨어 이외의 다른 많은 분야에서 사용했던 심리적인 방법들에 대해서 알아보고, 이들이 소프트웨어 실무에 적용하기 위해서는 어떠한 점을 고려하여 적용하여야 하는지 살펴보도록 하겠다.

### 3.1 멘토링(Mentoring)

멘토링이란 풍부한 경험과 전문지식을 가진 사람이 상대적으로 그렇지 못한 사람에게 1:1로 전담하여 인지적, 심리적 도움을 제공하는 활동으로써 일정 기간 동안 상호간의 인품과 역량을 개발하는 인재육성 인간 관계 활동을 일컫는다[7]. 한국멘토링코칭센터에 따르면 멘토링의 기원은 ‘텔레마쿠스의 스승(보호자)’ 멘토(Mentor)란 호메로스의 ‘오디세이아(Odyssey)’에 나오는 오디세우스의 충실한 친구의 이름에서 유래하였다. 오디세우스가 트로이 전쟁에 출정하면서 아들 텔레마쿠스의 교육을 멘토에게 맡겼다. 멘토는 오디세우스가 전쟁에서 돌아오기까지 20여 년 동안 왕자 친구, 교사, 상담자, 때로는 아버지가 되어 그를 보살피 주었다는데서 유래되었다. 멘토란 조직 내에서 해당 직업 분야에 능통한 사람으로서 조언과 격려를 해주는 친구 더불어 다른 사람의 학습과 성장을 기꺼이 도와주는 사람을 의미하며, 멘티란 해당 분야의 경험은 부족하지만 충분한 잠재 능력과 개발 가능성을 지닌 사람을 의미한다.

멘토링의 기능은 [표 1]과 같이 기술전수를 통해 경력을 관리하는 기능과 심리 사회적 기능으로 크게 나눌 수 있다[7].

그리고 이러한 멘토링이 우리나라 기업들에 도입된 현황은 [표 2]와 같다[8].

[표 1] 멘토링의 기능

<b>경력관련 기능 (Career Functions)</b>	<ul style="list-style-type: none"> <li>· 후원 Sponsorship</li> <li>· 코칭 Coaching</li> <li>· 보호 Protection</li> <li>· 도전적 업무부여 Challenge</li> <li>· 노출/소개 Exposure/Visibility</li> </ul>
<b>심리사회적 기능 (Psychology Functions)</b>	<ul style="list-style-type: none"> <li>· 역할모델 Role Modeling</li> <li>· 상담 Counseling</li> <li>· 수용/지원 Acceptance/Confirmation</li> <li>· 우정 Friendship</li> </ul>

[표 2] 우리나라 멘토링 도입 기업 현황

구분	대상	방법
포스 테이터	신규직원	<ul style="list-style-type: none"> <li>· 멘토링 운영위원회를 통한 매칭과 지원</li> <li>- 멘토와 멘티를 매칭</li> <li>- 멘토링 활동비 지원</li> <li>· 멘토링 활동 적극 지원</li> </ul>
삼양사	신규직원	<ul style="list-style-type: none"> <li>· 멘토와 멘티를 지원하는 모니터제도 활용</li> <li>· 평가와 사후관리 활성화</li> <li>- 인터뷰, 홈페이지 등을 통한 평가로 우수커플 시상, 인사에 반영</li> </ul>
삼성 테크윈	신규직원 및 경력사원	<ul style="list-style-type: none"> <li>· 후견인제도를 공식화</li> <li>· 멘토링사무국이 단체 멘토링 및 홍보 주재</li> <li>- 개인간 멘토링 외에 등반, 운동 경기 등 합동 멘토링 강화</li> </ul>
YTN	신규직원	<ul style="list-style-type: none"> <li>· 단체영화관람</li> <li>· 멘토선정 시 상호업무성격이 다른 직종간 교차방식으로 선정</li> </ul>
조흥 은행	신규직원	<ul style="list-style-type: none"> <li>· 직접대면, 전화, 이메일 등</li> <li>· 연 2회 후견대상자 접촉 및 성과평가</li> <li>- 성과반영 및 포상</li> </ul>

그러면 다른 학문 분야에서는 이러한 멘토링 제도를 통해 심리적인 측면에서 효과가 있었던 사례가 있는지 살펴보자. 한 예로, 대학과 기업 간에 산학협력을 하는데 있어 멘토링을 통해 기업에서 산업체 전문가를 멘토로 지정하여 대학생들을 멘토링 해주었을 때, 멘토링의 효과에 대해 확인한 연구가 있다[9]. 이 연구에서는 멘토링을 통해 산학협력 프로젝트를 진행하는데 있어서 기업과 학교의 만족도가 높아지고, 학생의 전공 역량과 사회적응력이 높아짐을 입증하였다.

또한 교육대학원에서 연구된 사례로, 소수의 멘토와 다수의 멘티로 구성되어 소집단 팀을 조직하여 학습을 하는 팀별 조직학습이 이루어지는 집단에 대하여 협력학습멘토링을 시도하여 멘토링의 효과에 대해 확인한 연구가 있다[10]. 이 연구

에서는 협력학습멘토링을 과정을 통해 전문적 능력이나 역량 개발과 정보 공유, 그리고 상호 심리적 지지와 안정감의 획득 등 다양한 측면에서의 학습 성과를 얻고 있는 것을 확인하였다.

이처럼 다른 분야에서는 멘토링의 효과가 입증되어 있지만, 소프트웨어 분야에 적용을 하고자 할 때 다음과 같은 문제점들을 고려하여 적용하여야 한다.

- 멘토링 할 시간의 부족으로 인하여 형식적인 제도가 될 수 있다.
- 멘토, 멘티 당사자들이 어떠한 방법으로 접근해야 하는지를 인지하지 못하여 효과가 떨어질 수 있다.
- 멘토링 시 단순히 업무 기술 전수에만 치중되어 심리적인 기능을 하지 못하는 경우가 있다.

소프트웨어 프로젝트 구성원의 현실은 스스로 마음을 가진 존재로서 정당한 대우를 받고 있지 않고, 컴퓨터를 조종하는 기계로 취급된다[4]. 위와 같은 몇 가지만 주의를 한다면, 프로젝트를 수행하는 것도 결국 사람이 하는 일임을 인지하고, 프로젝트 구성원 개개인 이 기계가 아닌 마음을 가진 존재로 인식하여 구성원들을 이해함으로써 프로젝트의 성과가 향상될 것이다.

### 3.2 MBTI를 활용한 성격유형별 관리

한국 MBTI 연구소에 따르면, MBTI(Myers-Briggs Type Indicator)는 C.G.Jung의 심리유형론을 근거로 하여 Katharine Cook Briggs와 Isabel Briggs Myers가 보다 쉽고 일상생활에 유용하게 활용할 수 있도록 고안한 자기보고식 성격유형지표이다. 융의 심리유형론은 인간행동이 그 다양성으로 인해 종잡을 수 없는 것 같이 보여도, 사실은 아주 질서정연하고 일관된 경향이 있다는 데서 출발하였다. 그리고 인간행동의 다양성은 개인이 인식(Perception)하고 판단(Judgement)하는 특징이

다르기 때문이라고 보았다. MBTI는 인식과 판단에 의한 융의 심리적 기능이론, 그리고 인식과 판단의 향방을 결정짓는 융의 태도이론을 바탕으로 하여 제작되었다. 또한 개인이 쉽게 응답할 수 있는 자기보고(self report) 문항을 통해 인식하고 판단할 때의 각자 선호하는 경향을 찾고, 이러한 선호 경향들이 하나하나 또는 여러 개가 합쳐져서 인간의 행동에 어떠한 영향을 미치는가를 파악하여 실생활에 응용할 수 있도록 제작된 심리검사이다.

MBTI는 [그림 1]과 같이 4가지의 양극적 선호 경향으로 구성되어 있다. 선호경향이란 C.G.Jung의 심리 유형론에 따르면, 교육이나 환경의 영향을 받기 이전에 인간에게 잠재되어 있는 선천적 심리 경향을 말하며, 각 개인은 자신의 기질과 성향에 따라 아래의 4가지 양극지표에 따라 둘 중 하나의 범주에 속하게 된다.



[그림 1] MBTI의 4가지 선호경향

이 4가지 선호지표를 조합하여 만들어진 Myers와 Briggs가 고안한 16가지 성격유형을 한국 심리검사 연구소(KPTI : Korea Psychological Testing Institute)에서 표현한 바에 따르면 [그림 2]와 같다. 이 도표는 생각이 많은 내향성(I)은 도표의 위쪽 두 줄에, 적극적이고 활동적인 외향성(E)은 도표의 아래쪽 두 줄에, 감각형(S)은 도표의 왼쪽 두 줄에, 직관형(N)은 도표의 오른쪽 두 줄에 배치하였고, 분석적이고 논리적인 사고형(T)은 도표의 왼편과 오른편에 배치하고, 관계지향적인 감정형

(F)은 도표의 중앙에 배치시켰다. 또한 정리 정돈을 잘하는 판단형(J)은 도표의 아래 위로 배치하고, 개방적이며 때로는 즉흥적인 인식형(P)은 도표의 가운데로 모아놓았다.

ISTJ 세상의 소금형	ISFJ 임금뿔편 권력형	INFJ 예언자형	INTJ 과학자형
ISTP 백과사전형	ISFP 성인군자형	INFP 잔다르크형	INTP 아이디어 뱅크형
ESTP 수완좋은 활동가형	ESFP 사교적인 유형	ENFP 스파크형	ENTP 발명가형
ESTJ 사업가형	ESFJ 친선도모형	ENFJ 언변능숙형	ENTJ 지도자형

**그림 2] MBTI의 16가지 성격유형**

그럼 다른 분야에서는 사람 간의 문제가 생겼을 때 이러한 MBTI를 활용하여 개선이 된 사례가 있는지 살펴보자. 한 예로, 경영 혁신 프로젝트를 하는데 있어서 MBTI를 통해 팀원들의 성격유형을 파악하여 서로를 이해함으로써 팀워크 향상에 도움을 줄 수 있음을 입증하였다[11]. 팀원의 개별성에 대한 인정이 경영 혁신 프로젝트를 진행하는데 매우 중요함에도 이것이 간과됨으로서 서로에 대한 이해가 갈등으로 발전하고 상호 신뢰가 바탕이 되어야 하는 혁신 업무에 부정적인 영향을 미치고 있음을 확인하고, 이를 MBTI를 통해 효과적인 팀워크 분석과 상호 이해를 함으로써 프로젝트의 성과가 향상될 수 있음을 확인하였다.

또한 군대 조직 내에서 MBTI 검사 및 결과에 대한 피드백이 대인갈등 인식 정도의 변화에 어떤 영향을 주는지 실험을 통해 확인한 연구 사례가 있다[12]. 이 연구에서는 MBTI 성격유형 검사를

실시하고 이를 통해 구성원의 성격유형에 대한 이해의 기회를 부여하였으며 추가로 개별 MBTI 결과에 대한 피드백을 실시하여 조직 구성원 간의 갈등이 해소됨을 입증하였다. 하지만 이러한 방법을 바로 적용할 때 아래와 같은 점들은 주의를 하여야 한다.

- MBTI는 개인의 선호 경향이 원하는 방향으로 조사 되어 때때로 오류가 있을 수도 있다.
- MBTI를 통해 성격유형을 분류하게 됨으로써 사람에 대한 선입견을 심어줄 수 있고, 해당 성격유형의 방향으로만 육성시키려는 문제가 발생할 수도 있다.
- 자동화 툴의 부재로 인해 프로젝트에 참여하는 모든 사람들의 MBTI 유형을 파악하여 서로 이해를 하는 것이 어려울 수 있다.

소프트웨어 프로젝트도 다른 분야의 프로젝트와 마찬가지로, 프로젝트를 진행함에 있어 이해관계자가 많고 사람들 간의 이해와 신뢰가 바탕이 되어야 하는 분야이다. 위와 같은 몇 가지만 주의를 한다면, 사람들 간의 이해와 신뢰를 하기 위해 MBTI를 활용하여 이해관계자들 간의 성격을 파악하여 프로젝트를 진행한다면 팀워크 향상되어 프로젝트의 성과가 향상될 것이다.

**3.3 프로파간다(Propaganda)**

Wikipedia의 정의에 따르면 프로파간다는 ‘일정한 의도를 갖고 세론을 조작하여 사람들의 판단이나 행동을 특정 방향으로 이끌어 가는 것’으로 정의 되어 있다. 하지만 원천적인 정의는 ‘특정 견해나 방침과 관련해 대중의 지지를 끌어내기 위해 질서 정연하게 전개되는 운동’으로 긍정적인 정의가 더욱 중요하다[13]. 프로파간다 기술은 주로 정치, 경제, 군사 분야에서 한정적으로 사용되던 기술로 현재는 상품의 광고와 마케팅, 회사의 PR 및 홍보에 사용되고 있다.

프로파간다 기술 중에서 많이 사용하는 캠페인은 사람들의 생각을 변화 시켜 '물건구매'를 유도하거나 '단체의 생각'을 전달하는데 많이 사용되고 있다. 이와 관련하여 다른 분야에서 활용되고 있는 예를 보면 다음과 같다. 마케팅으로 유명한 대표적인 P&G 기업의 "Thank You Mom" 캠페인 전략은 수년간 해당 기업에서 지속적으로 진행해 오던 캠페인 전략이다. 2010년 벤쿠버 동계 올림픽 당시 '선수들도 그들의 엄마에게는 아직도 아들, 딸일 뿐이다'라는 슬로건으로 기업의 신뢰도를 높이는 방법으로 선전 기술을 사용하였다. 또 다른 예로, 나이키의 'Just Do it' 캠페인은 운동화의 판매 전략에서 라이프 스타일을 판매 하는 전략으로 변경하고 하는 캠페인으로 유명하다. 이 캠페인은 운동선수의 성취와 활동에 주목했고 나이키만의 마인드를 만드는 계기가 되었다.

소프트웨어 조직 차원에서도 프로파간다 기술이 쓰이지 않는 것은 아니다. 회사나 조직 차원에서 사용되는 간략한 예를 세 가지만 보겠다.

첫째로, 슬로건(Slogan)은 대중에게 특정한 행동을 유발시키는 선전 문구를 말한다. 슬로건은 간결하고 반복성이 있으며 시의 적절하게 외치기 쉬워야 한다[14]. 대표적인 예로 Google의 "Don't be evil"과 Apple의 "Think Different", 썬 마이크로 시스템즈의 "The network is the computer"가 있다. 이런 문구를 통해 고객들에게 각 회사의 이미지를 전달하고 소프트웨어 개발자도 그들의 조직문화에 적용하여 행동하게 된다.

둘째로, 부화뇌동(The Bandwagon)기법은 모든 사람, 적어도 우리들 모두가 그것을 하고 있다는 생각을 주려는 것이 목적으로 소속된 집단이 이 프로그램을 적용하고 있기 때문에 우세한 쪽에 있을 수 있다는 생각을 심어 주는 기술이다[14]. CMMI에서 성공한 기업이 쓴 개발 방법론을

모아서 작성한 이유는 개발 조직과 개발자를 좀 더 설득하기 쉽게 할 수 있기 때문이다. "Agile Work uses lean thinking" 논문의 주장내용도 Agile 개발 방법론에 '도요다'에서 성공한 "Lean thinking"이 포함되어 있다는 내용으로 우세한 쪽에 Agile이 있다는 주장을 한 것 또한 이 기술의 한 예이다. 이런 예들은 소프트웨어 개발 환경에서 어렵지 않게 찾아 볼 수 있다.

마지막으로 미사여구(Glittering Generalities)를 사용하는 기법이다. 듣기 좋은 말로서 어떤 것을 연상 시키는데 사용하여 증거사실을 찾아보지 않고도 어떤 것을 받아들이거나 인정하게 만드는 방법이다[14]. "To improve product quality, you must improve process quality"라는 말은 의심할 여지가 잘 보이지 않는 문구이다. 하지만, 프로세스가 잘 정의되어 있다고 제품의 품질이 올라갈 것인가를 생각해 보면 항상 그런 것은 아니다 [2]. 이와 유사하게 Brooks가 주장한 "늦고 있는 프로젝트에 추가로 인원을 투입시키면 개발을 더욱 지연 시킬 뿐이다."라는 문구도 잘 생각해 보면, 추가 인원을 투입하여 일정 부분 줄일 수 있는 부분도 있다는 것이다.

프로파간다의 기술을 이용하여 소프트웨어 개발 조직의 문제점을 완화시킬 수 있는 방법으로 사용하면 좋겠다는 제안 의견도 있다[15]. 하지만, 소프트웨어 분야에서 이런 기술이 팀과 그 안에 있는 사람에 대해서는 잘 사용되지 않는 것 같다. 조직은 프로파간다와 같은 심리기술은 고객을 위해 사용되어야 하는 비싼 기술로 인식하고 팀 리더가 재량껏 사용할 수 있는 기술로 바라보고 있다. 개발자들의 마음을 연구하지 않는 이유는 개발자 심리에 대한 지속적인 연구와 여러 정보가 필요하고 정보를 모아서 수치화 할 수 있는 미디어가 한정 되어 있으며 개발자의 심리를 연구하는 것에

대한 정량적인 효과에 대해서도 아직 활발히 이뤄지지 않기 때문이다. 결국 이런 상황의 반복으로 인해서 개발자들의 업무 의지는 개인이 알아서 이끌어 내야 하는 것으로 치부 되고 커뮤니케이션 문제[1] 또한 팀에서 알아서 해결해야 하는 것으로 인식되는 상황이 되었다.

### 3.4 Social Media

심리기법은 "인간을 포함한 동물의 학습이 환경의 자극에 대한 반응이라 주장"하는 행동주의 심리학에서 모든 정신과정을 컴퓨터의 정보처리과정에 비유하여 연구하는 인지심리학으로 발전하였다. 개인 통신기계(smart phone, handphone)의 보급률이 2011년 100%를 넘어섰고 SNS와 Twitter로 대표되는 Social Computing이 유행하고 있다. Daumsoft에 따르면 Social Media에서 발생하는 데이터를 이용하여 소비자의 라이프 스타일과 트렌드를 분석하고 잠재고객의 니즈를 이해하여 브랜드의 인지도 향상과 위험요소를 제거하는데 Big Data Mining 마케팅 기술을 사용하고 있다고 하였다.

이런 상황에서 의미 있는 사건이 발생하였다. 독일 국영방송 ZDF에 따르면, 버락 오바마의 선거 전략 본부는 약 2억 5천만건의 유권자 프로필 데이터를 Big Data 분석과 마이크로 타겟팅을 통해 개인화된 메시지로 유권자와 정교하게 대화를 할 수 있는 'Voter Activation Network' 시스템을 개발하였다. 이를 통해 선거를 개인화 수준으로 끌어 올렸고 유권자의 심리를 정확히 이해하는 시도를 하였다. 또한 친구의 설득 효과가 더 좋다는 것과 관심주제는 개인이 신뢰할 수 있는 인물을 활용하는 것이 유리하다는 선전 전략도 같이 사용이 되었다. 이렇게 만들어진 정보를 SNS를 통해 유권자 자발적으로 확산시키는 효과를 얻어 내었다.

경제와 정치에서 사용되는 심리기법은 개발자의 두뇌 작업으로 이루어진 소프트웨어 개발자 조직에 적용할 수 있다. 개발자의 심리를 여러 도구를 이용하여 실시간으로 그들의 마음과 생각을 이해할 수 있다면, 개발자의 동기를 이끌어 내고 소통을 원활히 하며 소프트웨어 조직의 관리 문제를 해결 할 수 있는 유용한 방법으로 기대한다. 또한 솔직하지 못한 프로젝트 상태 보고 문제도 해결할 수 있는 좋은 아이디어를 제공할 것이다[1]. 물론 이론적으로 분석할 모델이 존재하지 않는다는 한계는 존재한다. 하지만, 개발자의 행동과 심리를 이해하고 충분한 데이터를 수집한다면 좀 더 발전된 소프트웨어 개발 조직을 만들어 낼 수 있을 것이다.

## 4. 결론

지금까지 소프트웨어 공학의 발전에 있어서 People, Process, Technology 측면에서 People이 많이 연구되지 않은 상황을 살펴보고, 이러한 사람과 관련하여 다른 학문 분야에서 활발히 연구되고 있는 4가지 심리 방법(멘토링, MBTI를 활용한 성격유형별 관리, 프로파간다, Big data를 활용한 심리분석)에 대해 알아보았다. 또한 그 방법들이 다른 학문 분야에서 효과적으로 사용되고 있는 사례를 살펴보고 소프트웨어 업계에서는 어떻게 사용하면 성공할 수 있는지에 대해 연구해 보았다.

본문에서 언급되었던, 소프트웨어 분야에 각 방법들을 적용할 때 주의해야 할 점들을 조심하여 각 방법을 소프트웨어 산업에 맞게 재해석을 하여 활용한다면, 다른 학문 분야에서 이미 적용되어 효과를 보았던 방법인 만큼 소프트웨어 산업에서도 '사람' 때문에 발생하는 문제들을

해결하여 소프트웨어 프로젝트의 성과가 향상될 것으로 판단된다.

## 참고문헌

- [1] Charette, R.N., "Why software fails?", *IEEE Spectrum vol42, No9*, pp.42-49, 2005
- [2] Naur, P. and Randell, B. (Eds.), "*Software Engineering : Report of a conference sponsored by the NATO Science Committee*", Garmisch, Germany. Brussels, Scientific Affairs Division, NATO, 1968
- [3] Rowan. T. C. "Psychological tests and selection of computer programmers", *Journal of the Association for Computing Machinery vol4, No3*, pp.348-353, 1957
- [4] Weinberg, G.M., "*The Psychology of Computer Programming*". New York, Van Nostrand Reinhold, 1971
- [5] F.P. Brooks, Jr. "*The Mythical Man-Month*", Addison-Wesley, 1972.
- [6] K. Schwaber and M. Beedle. "*Agile Software Development with SCRUM*". Prentice Hall, 1st edition, 2001.
- [7] Kram, Kathy E., "*Mentoring at work: Developmental relationships in organizational life.*", University Press of America, 1988
- [8] 이만기, "기업에서 멘토링 시스템 도입과 관련한 이론적 모델연구", *고용동향분석 I 2006년 1/4 분기*, pp.93~123, 2006
- [9] 이중만, "정보통신분야 산학협력 효과성에 대한 영향요인 분석 : IT멘토링 사례중심으로", *한국*

*콘텐츠학회논문지 제11권 제8호*, pp.342~352, 2011

- [10] 박귀영, "학습 동아리를 통한 협력학습멘토링 사례 연구", 아주대학교, 2009
- [11] 김승기, "성격유형과 프로젝트 팀워크관리", *Consulting Review vol.4*, p23~30, 2005
- [12] 김동욱, "MBTI 피드백을 통한 조직구성원간의 갈등해소", 인제대학교, 2009
- [13] Edward Bernays, "*프로파간다*", 공존, 2009
- [14] US Army, "Psychological Operations Field Manual No. 33-1", 1979
- [15] Warren Harrison, "Propaganda and Software Development", *IEEE Computer vol21, No5*, 2004

## 저 자 소 개



소 정 연

2004년 서울여자대학교 컴퓨터공학과(학사)  
 2004년~현재 (주)삼성전자 무선사업부 책임연구원  
 2012년~현재 KAIST 소프트웨어대학원 석사과정

<관심분야> 소프트웨어프로세스 평가, 심리학,  
 멘토링



최 창 진

2002년 인하대학교 수학과, 컴퓨터공학과(학사)  
 2003년~2012년 (주)팅크웨어 네비게이션연구소  
 2012년~ KAIST 소프트웨어대학원 석사과정

<관심분야> 소셜미디어, 클라우드컴퓨팅



한 승 범

2011년 학점은행 독학사 컴퓨터공학(학사)  
 1997년~2001년 미래산업연구소 소프트웨어 개발  
 연구원  
 2002년~현재 고영테크놀러지 연구소 소프트웨어  
 개발PM  
 2011년~현재 KAIST 소프트웨어대학원 석사과정

<관심분야> 소프트웨어프로세스 개선, P-CMM



민 상 윤

1993년 샌프란시스코 주립대학교 전산학과(학사),  
 Cum Laude

1997년 KAIST 전산학과(석사), 소프트웨어 공학  
 2003년 KAIST 전산학과(박사), 소프트웨어 공학  
 2000년~현재 (주)솔루션링크 대표이사

KAIST Software대학원 겸직교수

<관심분야> 소프트웨어 공학, 소프트웨어  
 프로세스, 소프트웨어 품질,  
 개발방법론, 소프트웨어 안전성

# 분기점 기반 트리 분석을 통한 소프트웨어 클러스터링 결과 비교<sup>†</sup>

(Comparison of Software Clustering using Split Based Tree Analysis)

엄재철<sup>‡</sup>  
(Jaechul Um)

이찬근<sup>‡</sup>  
(Chan-gun Lee)

**요약** 본 연구에서는 소프트웨어 아키텍처 복원을 위한 계층적 클러스터링(Clustering) 결과를 정량적으로 평가 할 수 있는 비교 메트릭(Metric)을 제시한다. 소프트웨어 클러스터링의 정량적 평가는 소프트웨어의 구조적 변화에 대한 이해를 돕는 척도를 제시하는 연구이다. 이를 위해 생물정보학에서 상호 유전 형질 분석에 사용하는 분기점(Split) 개념을 소프트웨어 아키텍처 분석에 적용한다.

**키워드** 소프트웨어 아키텍처, 분기점, 클러스터링

**Abstract** We propose a novel metric for quantitatively comparing different clustered results generated from software clustering algorithms. A quantitative evaluation of software clustering helps understanding of architectural changes of software. The concept of split, which has been used for analysis of genetic characters in bio-informatics, is applied in the analysis of software architecture.

**Key words** Software Architecture, Split, Clustering

## 1. 서론

소프트웨어 아키텍처는 소프트웨어 시스템 구조와 동작을 기술한 구성도로 소프트웨어 전반에 대한 이해, 개발, 유지보수 등에 도움을 준다. 일반적으로 소프트웨어 아키텍처는 소프트웨어가 진화함에 따라 변화하게 되는데, 이 때 설계 문서가 같이 수정되지 않는 문제가 발생할 수 있다. 이러한 문제는 소프트웨어 아키텍처 복원을 통해 해결될 수 있는데, 이것을 위한 주요한 방법이 소프트웨어 클러스터링이다. 본 논문에서는 소프트웨어 클러스터링 결과 비교 방법을 분기점 기반 트리

분석으로 제시한다. 이러한 소프트웨어 클러스터링 결과 비교는 소프트웨어 설계 분석, 역공학, 리팩토링 등을 돕는 방법으로 활용될 수 있다.

## 2. 관련 연구

기존의 소프트웨어 클러스터링 결과 비교를 위한 대표적인 방법으로는 MoJo[1]가 있다. MoJo는 소프트웨어 아키텍처를 여러 개의 집합으로 나누고 Move와 Join 연산을 통해 비교한다. MoJo는 특히 소프트웨어 역공학 및 클러스터링 분야에서 상호 소프트웨어 아키텍처를 비교하는데 주로 사용된다. 그 밖에도 MoJo를 보완한 UpMoJo[2], MoJo를 수식화하여 재정의한 MoJoFM[3] 등이 소프트웨어 클러스터링 결과 비교를 수행하고 있다.

<sup>†</sup> 본 연구는 한국연구재단 기초연구사업(과제번호 2010-0015636, 2011-0013924)의 지원을 받았습니다.

<sup>‡</sup> 학생회원 : 중앙대학교 컴퓨터공학부  
steveum0105@gmail.com

<sup>‡</sup> 종신회원 : 중앙대학교 컴퓨터공학부 교수. 교신저자  
cglee@cau.ac.kr

### 3. 소프트웨어 클러스터링 결과 비교

#### 3.1 분기점 탐색

분기점은 발생학에서 어떤 두 유전 형질에 대한 공통 부모를 의미한다. 이러한 분기점은 여러 방법으로 분석되어 유전 형질의 파생 정보를 나타내는 요소로 사용된다. T. Mailund[4]와 Q. Zhang[5]은 분기점으로 트리 구조를 분석하는 방법을 설명하고 있다. T. Mailund는 트리에서 분기점을 찾고, 분기점 간의 거리를 측정하여 서로 다른 구조의 트리 집합을 비교할 수 있음을 설명하고 있다. Q. Zhang은 분기점을 이용한 트리 분석을 위한 새로운 메트릭을 제시하고 그것을 클러스터링에 응용할 수 있음을 보인다.

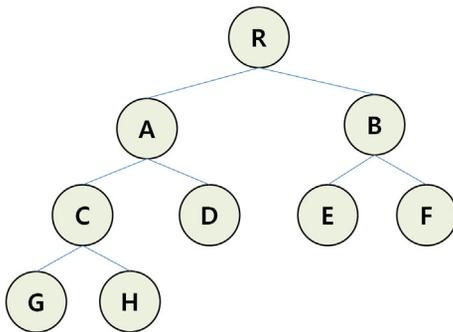


그림1. 유전 트리

그림1은 어떤 생물의 발생학적 유전 트리를 나타낸다. 그림1에서 개체 G와 D가 어떤 공통된 형질을 가지고 있다면, 그것은 공통된 부모인 A에서 유전되었다고 할 수 있다. 이 때 A는 G와 D의 분기점이며,  $A = \text{Split}(G, D)$ 와 같이 표현된다. 다음 식은 분기점의 일반 정의를 나타낸다.

$$N_s = \text{Split}(n_i, n_j) \quad (1)$$

(1)은 두 노드  $n_i, n_j$ 에 대한 분기점 노드  $N_s$ 를 정의한다. 소프트웨어 아키텍처에서 분기점 역시

(1)의 정의를 따른다. 이 때 소프트웨어 아키텍처를 나타내는 트리는 클래스 단위 노드(Node)와 상속 관계의 선분(Edge), 파일 단위 노드와 패키지 구성 관계의 선분, 모듈 단위 노드와 의존성(Dependency) 관계의 선분 등 다양하게 표현될 수 있다.

본 논문에서는 소프트웨어 아키텍처의 계층 구조에서 분기점을 탐색한 후, 그것을 분석하여 소프트웨어 클러스터링 결과 비교를 수행한다.

#### 3.2 분기점 순서(Split-Order)를 이용한 클러스터링 결과 비교

Q. Zhang은 일반적인 클러스터링 결과 비교를 위해 분기점 순서 방법을 제시한다. 분기점 순서는 각 클러스터링 결과에 대해 분기점을 탐색한 후, 그 분기점과 루트 간의 거리를 대소 관계로 비교한다.

세 노드  $\alpha, \beta, \gamma$ 에 대한 분기점 순서는  $\text{SplitOrder}(\alpha, \beta, \gamma)$ 와 같이 표현하며 다음과 같이 정의된다.

- $\text{SplitOrder}(\alpha, \beta, \gamma) = '<'$ ,  $\text{Split}(\alpha, \beta)$ 가  $\text{Split}(\alpha, \gamma)$ 보다 상위 레벨 노드일 때
- $\text{SplitOrder}(\alpha, \beta, \gamma) = '>'$ ,  $\text{Split}(\alpha, \gamma)$ 가  $\text{Split}(\alpha, \beta)$ 보다 상위 레벨 노드일 때
- $\text{SplitOrder}(\alpha, \beta, \gamma) = '='$ ,  $\text{Split}(\alpha, \beta)$ 와  $\text{Split}(\alpha, \gamma)$ 가 동일 레벨 노드일 때

그림2, 3은 계층적 클러스터링의 서로 다른 결과이다. 위의 그림2, 3에서 모든 노드 쌍에 대해 분기점 순서를 구하고 서로 같은 값을 갖는 노드 쌍의 개수가 유사도가 된다.

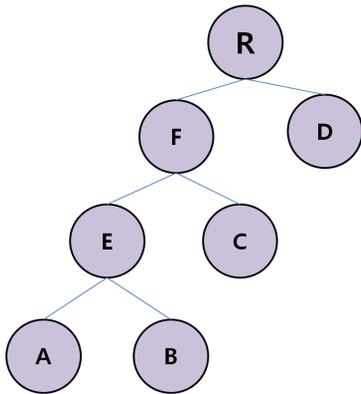


그림2. 계층적 클러스터링 결과 A

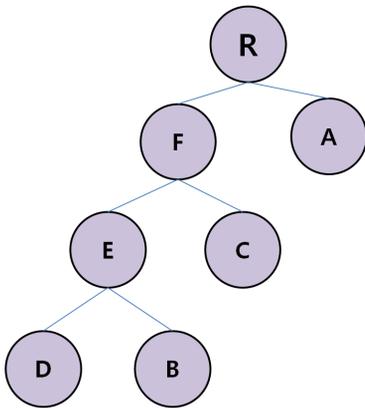


그림3. 계층적 클러스터링 결과 B

그러나 Q. Zhang의 분기점 순서 방법은 일부 경우에 직관적이지 않은 결과를 나타낼 수 있다. 가령 그림2와 3은 노드 A와 D의 위치가 바뀐 것을 제외하고는 서로 동일한 클러스터링 결과이다. 분기점 순서 방법을 이용한 리프노드 A~D에 대한 비교에서 총 12개의 조합쌍 중 서로 같은 기호의 개수는 0개이다. 즉, 그림2와 3은 서로 어느 정도 유사한 부분이 있음에도 유사도가 0으로 출력된다.

3.3절에서는 분기점 순서의 이러한 문제점을 개선한 계층적 클러스터링 결과 비교 방법을 제시하고, 소프트웨어를 대상으로 한 비교 분석을 논의한다.

### 3.3 분기점을 이용한 소프트웨어 클러스터링 결과 비교

3.1절에서 소개한 분기점의 정의를 이용하여 동일 소프트웨어의 클러스터링 결과를 비교한다. 이를 위해 다음 식을 사용한다.

$$D = \sum_{n_i, n_j \in N}^N |\text{dist}(R_2, \text{Split}(n_i, n_j)) - \text{dist}(R_1, \text{Split}(n_i, n_j))| \quad (2)$$

$$\text{MaxD} = \sum_{n_i, n_j \in N}^N \max(\text{dist}(R_2, \text{Split}(n_i, n_j)), \text{dist}(R_1, \text{Split}(n_i, n_j))) \quad (3)$$

$$d = \frac{\text{MaxD} - D}{\text{MaxD}} \quad (4)$$

(2)는 서로 다른 두 클러스터링 결과의 모든 노드  $N$ 에 대하여 분기점과 루트  $R_1$ , 분기점과 루트  $R_2$  간의 거리 차의 합을 정의한다. 이 때  $\text{dist}(\alpha, \beta)$ 는 두 노드  $\alpha$ 와  $\beta$  간의 최단거리를 의미하고,  $D$ 는 두 클러스터링 결과 간의 차이를 의미한다. (3)은 두 클러스터링 결과에서 같은 노드 쌍에 대해 두 분기점과 루트 거리 중 최댓값을 합한 것으로  $D$ 가 가질 수 있는 최댓값을 의미한다. (4)는 두 클러스터링 간의 유사도  $d$ 를 구하기 위해  $\text{MaxD}$ 에 대한  $D$ 의 비율을 정의한다.

그림 2와 3에 대해  $D$ 는 6이며,  $\text{MaxD}$ 는 7이다. 따라서 두 클러스터링 결과의 유사도  $d$ 는 7분의 1로 약 14%가 된다. 이것은 앞선 3.2장에서 분기점 순서의 유사도가 12분의 0으로 0%가 나왔던 것과 다르게 클러스터링 결과가 일부 유사함을 반영하고 있음을 보인다.

## 4. 결 론

본 논문에서는 소프트웨어 클러스터링 결과 비교를 위해 분기점을 이용한 새로운 트리 분석 방법을 제시했다. 현재 본 논문에서 제시하는 메트릭을 소프트웨어 설계 분석, 역공학 등 다양한 분야에 활용하는 방법을 확장 연구하고 있으며, 기존의 다른 방법론과의 비교 분석 또한 연구를 계속하고 있다.

## 참 고 문 헌

- [ 1 ] V. Tzerpos and R. C. Holt, "MoJo: A Distance Metric for Software Clustering," Proc. of Working Conference on Reverse Engineering, 1999.
- [ 2 ] M. Shtern and V. Tzerpos, "Lossless Comparison of Nested Software Decompositions," Proc. of Working Conference on Reverse Engineering, 2007.
- [ 3 ] Z. Wen and V. Tzerpos, "An effectiveness measure for software clustering algorithms," Proc. of the 12th IEEE International Workshop on Program Comprehension, 2004.
- [ 4 ] T. Mailund, "SplitDist-Calculating Split-Distances for Sets of Trees", Proc. of BiRC, 2004.
- [ 5 ] Q. Zhang, E. Y. Liu, and W. Wang, "Split-Order Distance for Clustering and Classification Hierarchies", Scientific and Statistical Database Management, Vol.5566, pp.517-534, 2009.

## 저자소개



엄 재 철

2012년 중앙대학교 컴퓨터공학부 학사.  
2012년~현재 중앙대학교 컴퓨터공학부 석사과정.

<관심분야> 실시간 소프트웨어, 소프트웨어 클러스터링, 데이터 마이닝



이 찬 근

1996년 중앙대학교 전자계산학과 학사  
1998년 KAIST 전산학과 석사.  
2005년 Univ. of Texas at Austin 전산학과 박사.  
2005년~2007년 미국 인텔 소프트웨어 엔지니어.  
2007~현재 중앙대학교 컴퓨터공학부 조교수.

<관심분야> 실시간 소프트웨어, 수행시간 모니터링, 소프트웨어 테스트



## 회원 가입 안내 및 회비 납부 요령



한국정보과학회 소프트웨어공학소사이어티는 회원 여러분에게 유익한 정보를 제공해 드리기 위하여 보다 충실한 내용의 논문지 발간 배포, 그리고 국제·국내 학술발표회 및 초청강연회와 단기강좌 등의 여러 가지 사업들을 추진하고 있습니다.

소프트웨어공학 소사이어티의 가입을 통해 정보 및 기술 교류, 그리고 인적 네트워크의 구성에 참여하시기를 기대합니다. 회원 가입을 위하여 아래의 회비 안내를 참고하시어 회비를 납부하시고, 다음 쪽의 입회원서를 작성하시어 아래 소프트웨어공학소사이어티 주소로 보내주시거나 팩스 또는 이메일을 통해 보내어 주시기 바랍니다.

한국정보과학회 소프트웨어공학소사이어티 연락처는 아래와 같습니다.

### ◆ 소프트웨어공학소사이어티

주 소 : (우) 121-742 서울시 마포구 신수동 1번지, 서강대학교 신과학관 202호  
한국정보과학회 소프트웨어공학소사이어티 박수용

전 화 : (02) 705-8928

팩 스 : (02) 704-8273

전자우편 : [sypark@sogang.ac.kr](mailto:sypark@sogang.ac.kr) (박수용교수), [bjlee@uos.ac.kr](mailto:bjlee@uos.ac.kr) (이병정교수)

홈페이지 : <http://www.sigse-kiss.or.kr/>

### ◆ 회비 안내

회원구분	<ul style="list-style-type: none"> <li>• 학생회원 : IT 분야 학과 또는 관심 있는 학생</li> <li>• 정회원, 종신회원 : IT 분야 종사자</li> </ul>
가입비	학생회원, 정회원 : 20,000원, 종신회원 : 200,000원
년회비	학생회원, 정회원 : 20,000원

### - 회비납부 방법

- (1) 무통장입금 또는 계좌이체 후 입회원서 발송  
: 계좌 번호 : 제일은행 150-20-358028 (이병정)
- (2) 소사이어티 주관 학술행사 개최시, 행사장 당일 가입 및 납부 가능



# 개인회원용 입회원서



회원구분	학생회원 ( ) 정회원( ) 종신회원( )				
성명	한글		생년월일		
	영문				
연락처	직장전화		휴대전화		
	e-mail				
주소	직장명/ 부서		직급		
	직장주소	(우)			
학력	학사	년 월 - 년 월	대학교	과	
	석사	년 월 - 년 월	대학원	과	
	박사	년 월 - 년 월	대학원	과	
관심분야					

본인은 한국정보과학회 소프트웨어공학소사이어티의 취지에 찬성하여 회원으로 가입하고자 이에 입회원서를 제출합니다.

년 월 일

신청인: (인)

한국정보과학회 소프트웨어공학 소사이어티 회장 귀하



## 논문지 논문 모집 (Call for Papers)



한국정보과학회 소프트웨어공학 소사이어티에서는 매년 4회에 걸쳐 ‘소프트웨어공학 소사이어티 논문지’를 발간하고 있습니다. 이 논문지에는 소프트웨어공학 전반에 걸친 연구논문과 산업계 논문을 게재해 오고 있습니다. 다음과 같은 소프트웨어공학 주제에 관련된 논문을 모집하고 있으니 학계와 산업계의 여러분의 적극적인 논문투고를 바랍니다.

### ◆ 논문 주제

- 소프트웨어 설계 및 아키텍처
- 소프트웨어 재사용 및 프로덕트라인
- 요구공학
- 소프트웨어 품질 및 테스트
- 관리 및 프로세스
- 소프트웨어 정형 기법
- 서비스기반 소프트웨어 개발
- 임베디드, 모바일, 웹기반 소프트웨어 개발
- 기타 소프트웨어 응용 (국방, 자동차, 조선 등의 분야)

### ◆ 논문심사

- 투고된 논문은 편집위원회에서 심사 선정하며, 필요 시 외부 심사위원을 위촉하여 심사를 합니다. 제출된 논문은 반환하지 않습니다.
- 심사료 및 게재료: 없음

### ◆ 논문 제출

- 소프트웨어공학 소사이어티의 논문지 투고 양식(<http://www.sigse-kiss.or.kr/>)을 사용하며, 논문의 분량은 10장으로 제한합니다.
- 논문지 투고규정에 따라 작성된 심사용 논문파일은 온라인투고시스템을 통하여 투고하시기 바랍니다.

### ◆ 문의처 (편집위원회)

- 편집이사 : 강성원 교수 (KAIST, 042-350-3512, sungwon.kang@kaist.ac.kr)
- 편집이사 : 윤희진 교수 (협성대학교, 031-299-0841, hgyoon@uhs.ac.kr)
- 편집이사 : 이관우 교수 (한성대학교, 02-760-5864, kwlee@hansung.ac.kr)
- 편집이사 : 채홍석 교수 (부산대학교, 051-510-3517, hscha@pusan.ac.kr)
- 편집이사 : 김문주 교수 (KAIST, 042-350-3543, moonzoo@cs.kaist.ac.kr)
- 편집위원 : 김정아 교수 (관동대학교, 033-649-7801, clara@kwandong.ac.kr)
- 편집위원 : 김현수 교수 (충남대학교, 042-821-6657, hskim401@cnu.ac.kr)
- 편집위원 : 이우진 교수 (경북대학교, 053-950-6378, woojin@mail.knu.ac.kr)
- 편집위원 : 박수진 교수 (서강대학교, psjdream@sogang.ac.kr)
- 편집위원 : 이지현 교수 (대전대학교, jihyun30@dju.ac.kr)
- 편집위원 : 최종무 교수 (단국대학교, choijm@dankook.ac.kr)
- 편집위원 : 김태호 박사 (ETRI, taehokim@etri.re.kr)



## 투 고 요 령



1. 소프트웨어공학소사이어티 논문지에 실리는 원고는 주제 논문, 일반 논문, 산업체 기고 등으로 구분하며 다음과 같은 분야에 대하여 모집한다.
  - 가. 소프트웨어공학 및 그 응용분야에 대한 연구결과
  - 나. 강좌 및 관련 교육사항 소개 (목적, 과정, 일정, 대상, 특징)
  - 다. 소프트웨어 도구 및 방법론 소개 (가격, 특징, 종류, 적용사례)
  - 라. 소프트웨어 산업에 대한 학계, 업계의 주요 관심사
  - 마. 기타 관련 사항
2. 투고자는 원칙적으로 본 소사이어티의 회원으로 한다. 다만 공동 또는 초청 기고자는 예외로 한다.
3. 논문은 원칙적으로 한글로 작성한다.
4. 원고는 한글(hwp), 워드(MS Word), PDF 형식 중 하나를 택하여 A4용지에 작성하며, 그림과 표를 포함하여 10쪽 이내로 한다.
5. 논문 내용에 직접 관련이 있는 문헌에 대해서는 이들 문헌에 관련이 있는 본문 중에 참고 문헌 번호를 쓰고 그 문헌을 참고문헌 난에 인용 순서대로 기술한다. 참고문헌은 학술지의 경우 저자, 제목, 학술지명, 권, 호, 쪽수, 발행 연도의 순서로, 단행본은 저자, 서명, 쪽수, 발행처, 발행 연도의 순서로 기술한다.

[1]Cole, R., "Parallel Merge Sort", SIAM Journal of Computing, vol.17, No.4, pp.770-785, 1988.  
[2]김수형, 강명호, 조형재, 송주석. "안전하고 효율적인 침입자 역추적 시스템", 정보과학회논문지, 제25권, 제10호, pp.1123-1131, 1998
6. 논문은 소프트웨어공학 소사이어티(<http://www.sigse-kiss.or.kr/>)의 온라인 투고 시스템을 통해 제출한다.
7. 논문투고신청서를 반드시 작성하여 이메일([hjyoon@uhs.ac.kr](mailto:hjyoon@uhs.ac.kr))로 제출한다.
7. 원고 접수는 수시로 하며, 접수일은 온라인 접수일로 한다.
8. 기타 자세한 사항은 한국정보과학회 논문지 투고 요령을 따른다.



## 한국정보과학회 소프트웨어공학소사이어티 임원명단

구분	성명	소속기관명	E-Mail	
회장	한혁수	상명대학교	hshan@smu.ac.kr	
부회장 (기획)	권기현	경기대학교	khkwon@kyonggi.ac.kr	
부회장 (편집)	강성원	KAIST	sungwon.kang@kaist.ac.kr	
부회장 (학술)	홍장의	충북대학교	jehong@chungbuk.ac.kr	
부회장 (조직)	이병걸	서울여자대학교	byongl@swu.ac.kr	
부회장 (협력)	전진옥	비트컴퓨터	jojeon@bit.co.kr	
운영위원회	총무이사	이정원	아주대학교	jungwony@ajou.ac.kr
		유준범	건국대학교	jbyoo@konkuk.ac.kr
	기획이사	이병정	서울시립대학교	bjlee@uos.ac.kr
		서주영	아주대학교	jyseo@ajou.ac.kr
	조직이사	백종문	KAIST	jbaik@kaist.ac.kr
		김영철	홍익대학교	bob@hongik.ac.kr
	학술이사	인호	고려대학교	hoh_in@korea.ac.kr
		고인영	KAIST	iko@kaist.ac.kr
	편집이사	윤회진	협성대학교	hjyoon@uhs.ac.kr
		이관우	한성대학교	kwlee@hansung.ac.kr
		채홍석	부산대학교	hschae@pusan.ac.kr
		김문주	KAIST	moonzoo@cs.kaist.ac.kr
	홍보이사	조은숙	서일대학교	escho@seoil.ac.kr
		이찬근	중앙대학교	cglee@cau.ac.kr
		박용범	단국대학교	ybpark@dankook.ac.kr
	협력이사	이세영	NIPA	sarahlee230@gmail.com
감사	차성덕	고려대학교	scha@korea.ac.kr	
	이상은	NIPA	selee@nipa.kr	
자문위원회	강교철	포항공과대학교	kck@postech.ac.kr	
	권용래	KAIST	kwon@cs.kaist.ac.kr	
	성기수	KISTI 고문	Kss13@truefriend.com	
	배두환	KAIST	bae@se.kaist.ac.kr	
	신규상	ETRI	gsshin@etri.re.kr	
	양승민	송실대학교	smyang@ssu.ac.kr	
	우치수	서울대학교	wuchisu@selab.snu.ac.kr	
	이경환	중앙대학교	kwlee@object.cau.ac.kr	
	이단형	KAIST	danlee@cs.kaist.ac.kr	
	정기원	송실대학교	chong@comp.ssu.ac.kr	
	박수용	서강대학교	sypark@sogang.ac.kr	
	황선명	대전대학교	sunhwang@dju.kr	
	전진옥	비트컴퓨터	jojeon@bit.co.kr	
	김수동	송실대학교	sdkim777@gmail.com	
	이궁해	항공대학교	khlee@kau.ac.kr	
최병주	이화여자대학교	bjchoi@ewha.ac.kr		

구분	성명	소속기관명	E-Mail
이사	김정아	관동대학교	clara@kwandong.ac.kr
	남영광	연세대학교	yknam@yonsei.ac.kr
	박수진	서강대학교	psjdream@sogang.ac.kr
	염근혁	부산대학교	yeom@pusan.ac.kr
	오재원	카톨릭대학교	jwoh@catholic.ac.kr
	윤희병	국방대학원	hbyoon37@hanmail.net
	이우진	경북대학교	woojin@knu.ac.kr
	이은석	성균관대학교	leees@skku.edu
	이석원	아주대학교	leesw@ajou.ac.kr
	이은주	경북대학교	ejlee@knu.ac.kr
	전태웅	고려대학교	jeon@korea.ac.kr
	정인상	한성대학교	insang@hansung.ac.kr
	최승훈	덕성여자대학교	csh@duksung.ac.kr
	최종무	단국대학교	choijm@dankook.ac.kr
	최호진	KAIST	hojinc@kaist.ac.kr
	현창문	탐라대학교	cmhyun@tnu.ac.kr
	계승교	삼성SDS	seankae@samsung.com
	권경룡	국방기술품질원	ka-ja17@hanmail.net
	권원일	STA컨설팅	wonil@softwaretesting.co.kr
	김상기	현대자동차	sangkikim@hyundai-motor.com
	김진태	SEEG	jtkim@swexpertgroup.com
	민경오	LG전자	davidmin@lge.com
	민상윤	솔루션링크	sang@sol-link.com
	박복남	핸디피엠지	pbnknb@hitel.net
	박찬규	국방SW산학연합회	milspark@hotmail.com
	배현섭	슈어소프트테크	hsbae@suresofttech.com
	손세창	인천공항공사	scsohn@airport.kr
	손진규	삼성탈레스	Jinkyu.son@samsung.com
	신석규	SW시험인증센터	skshin@tta.or.kr
	양상욱	KAI	sangyang@koreaaero.com
	유영수	현대엠앤소프트	ysyoo@hyundai-mnsoft.com
	윤태권	한국SW기술진흥협회	tkyune@empal.com
	윤형진	케피코	Hyoungjin.yoon@kefico.co.kr
	이근	삼성전자	gskeun.lee@samsung.com
	이성남	방위사업청	dapalee@korea.kr
	이우복	삼성전자	woobok.yi@samsung.com
	이장수	한국원자력연구원	jslee@kaeri.re.kr
	장주수	모아소프트	jsjang@moasoftware.co.kr
	정연대	N3SOFT	ydchung@n3soft.co.kr
	조병인	국방과학연구소	chobyun@dreamwiz.com
	조상윤	다한테크	sycho@dahan.co.kr



## 2011-2012 소프트웨어공학소사이어티 논문지 편집위원회

편집위원장 강성원 교수(KAIST)

편집위원 김문주 교수(KAIST)

김정아 교수(관동대학교)

김현수 교수(충남대학교)

박수진 교수(서강대학교)

윤회진 교수(협성대학교)

이관우 교수(한성대학교)

이우진 교수(경북대학교)

이지현 교수(대전대학교)

채흥석 교수(부산대학교)

최종무 교수(단국대학교)

김태호 박사(ETRI)



## 소프트웨어공학소사이어티 논문지 제25권 제3호 (통권 95호)

발행일 || 2012년 9월 30일

발행인 || 한혁수

편집인 || 강성원

발행처 || 사단법인 한국정보과학회 소프트웨어공학소사이어티

연락처 || 서울특별시 종로구 홍지문 2길 20, 상명대학교 소프트웨어 대학관 G511

전화 : 02-2287-5033, 팩스 : 02-2287-0049

홈페이지 : <http://www.sigse-kiss.or.kr/>

인쇄처 || (주)참기획 (전화 : 042-861-6380, 팩스 : 042-861-6381)

Copyright© 2012 한국정보과학회 소프트웨어공학소사이어티(비매품)