

---

# 소프트웨어 개발인력 배치를 위한 수학적 업무 배정 방법

진상\*, 이상준\*\*, 서성채\*, 김병기\*

## A Mathematical Approach of Work Assignment for Human Resource in Software Development

Xiang Chen\*, Sang-Joon Lee\*\*, Seong-Chae Seo\*, Byung-Ki Kim\*

**요약** 애자일 개발방법론 기반의 소프트웨어 개발 프로젝트에서도 각광받고 있는 팀의 공동작업이 일반화되고 있다. 효율적인 팀워크는 보다 효과적인 팀 관리운영과 프로젝트의 성과 측면에서 매우 중요하다. 기존의 소프트웨어 개발 인력배치는 휴리스틱 방법을 사용하였지만 이를 보완하기 위한 알고리즘적 접근법도 필요하다. 본 논문에서는 팀워크 기반의 소프트웨어 개발 프로젝트에서 개발자의 인력 배치를 위한 수학적 접근 방안을 제안하였다. 제안한 방안은 6개의 프로세스로 구성되며, 각 프로세서에서의 활동을 수학적으로 정의하였고, 기능배치매트릭스를 사용하였다. 본 논문에서 제시한 접근방법의 유용성을 증명하기 위하여 사례 연구를 제시하였다. 본 논문은 그동안 직관적이거나 경험적 방법으로 팀을 구성하는 방식에서 벗어나, 계량적인 논리에 의해 인적자원을 할당하는 수학적 접근방법을 개발하였다는데 큰 의의가 있다.

**주제어** : 인적자원 관리, 프로젝트 관리, 수학적 모델, 기능적 배치 매트릭스, 사례연구

**Abstract** Team collaboration is becoming commonplace and it is spotlighted in agile software development projects as well. More efficient teamwork in terms of effective team operation and project performance is very important. Heuristic software development staffing method has been used, but algorithm approach is needed to compensate for it. In this paper, we propose a mathematical approaches for staffing developers in teamwork-based software development projects. This consist of six process, and activities in each processor is defined as a mathematical function placement, and functional deployment matrix is used. A case study is presented in order to prove the usefulness of this approach. This paper is a significant research because a mathematical approach of work assignment is developed for human resources by quantitative logic and it deviate from intuitive or heuristic methods used previously.

**Key Words** : Human Resource Management, Project Management, Mathematical Model, Functional Deployment Matrix, Case Study

---

### 1. 서론

소프트웨어 개발 프로젝트는 업무 특성상 참여인력이 생산성과 산출물에 미치는 영향이 매우 크다. 프로젝트에 참여하는 개인들 즉 팀원들이 가진 지식과 능력에 의해 프로젝트 결과물에 많은 영향을 미친다. 그렇기 때문에 인적 자원 관리가 그 어떤 분야보다 중요하다[1].

프로젝트 관리는 집단 속에서 함께 일하는 개인들이

정해진 목표를 효율적으로 달성할 수 있도록 환경을 조성하고 유지해 나가는 과정이고 계획과 표준에 따라 비용, 일정, 인력, 위험을 통제하고 평가하여 사업을 성공적으로 수행하도록 한다[15]. 소프트웨어 프로젝트 관리는 계획을 수립하고, 조직화 하며, 방향을 제시하고, 조직의 자원을 통제하는 일이고 구체적인 목적과 목표를 완수하기 위하여 설정한 비교적 단기의 목표를 달성하려는 것

---

\*전남대학교 전자컴퓨터공학과

\*\*전남대학교 경영학부(교신저자)

논문접수: 2012년 12월 29일, 1차 수정을 거쳐, 심사완료: 2013년 1월 20일, 확정일: 2013년 2월 20일

이다. 그리고 수직적 계층의 기능조직의 인력들을 수평적 계층의 특정 프로젝트에 할당하는 시스템적 접근방법을 관리에 적용하려는 것이다[11].

날로 복잡해지는 사용자의 요구와 개발기술의 급속한 발전으로 소프트웨어의 체계적인 관리의 필요성이 증대되고 있으며, 소프트웨어 개발은 개발자들이 팀을 이루어 계획과 통제에 따라 진행시키는 형태로 이루어지고 있다[17]. 소프트웨어 개발 현장에서, 짧은 주기의 반복적이고 지속적인 제품 출시를 통해 시장의 반응을 빠르게 적용하며, 전문성이 높은 소규모의 팀을 구성하여 효과적인 대화와 협력으로 생산성을 극대화 하는 애자일 방법론이 널리 적용되고 있다[6,19]. 소프트웨어 개발 환경에서 애자일방법론이 빠른 속도로 전 세계에 확산되고 있다[7,8]. 포레스트 리서치의 2008년 2월 보고서[18]에 따르면, 북미와 유럽 지역 기업의 25% 이상이 공식적으로 애자일 방법론을 도입했으며, 나머지 기업 가운데 50%가 이미 도입 계획을 가지고 있거나 검토 중에 있다. 대표적인 애자일 방법론인 XP나 Scrum에서는 인간의 요소를 강조하고 있으며, 개인 요인뿐만 아니라 팀워크를 중요하게 다루고 있다[17].

팀워크를 갖추기 위해서 가장 먼저 해결해야 할 문제는 프로젝트 초기에 작업에 합당한 개발자를 할당하는 것이다. 특히 프로젝트 규모가 커지면 개발자가 점점 많아지게 되어 작업과 인원 분배의 복잡도가 커지게 된다. PMBOK Guide[15]에서는 작업에 가장 적합한 능력을 갖춘 인적 자원을 할당하는 것은 개발의 효율성을 증진한다고 언급하고 있지만, 작업과 인적 자원 할당에 대한 논의는 거의 없는 실정이다.

본 논문에서는 애자일 개발방법론 등을 채택한 소프트웨어 개발 프로젝트에서 작업과 개발자의 분배 문제점을 극복하기 위해 팀워크 개념을 추가한 방법을 제안한다. 제안된 모델은 작업과 인적자원 분배를 위해 수학적 모델을 만들고 6-시그마의 FDM 우선순위 계산[2]과 거리계산 등을 적용해서 작업에 가장 합당한 사람을 할당한다. 이를 통해 필수적으로 팀워크를 유지하는 팀워크 기반의 애자일 개발방법의 효율성을 재고하고자 한다.

본 논문에서는 2장에서 관련연구, 3장에서 수학적 배정 방법, 4장에서는 사례연구를 소개하고, 5장에서는 결론과 향후 연구 방향을 논의하였다.

## 2. 관련연구

소프트웨어 관리자에게는 할당된 인력 (비용)을 낭비 없이 적시에 효율적으로 활용하여 주어진 기간 내에 개발을 완료하는 능력이 필수적으로 요구된다[3]. 소프트웨어 개발 계획에 의거하여 개발기간 내에 총 소요 인력을 적절히 분배할 수 있어야 하며, 이때 사용되는 개발인력 분포를 인력 프로파일 (Staffing Profile)이라 한다.

인력 프로파일은 시간단위 설정 방법에 따라 단위 시간 (일, 주간, 월)으로 설정하는 시간기반 방법과 소프트웨어 생명주기의 단계로 설정하는 단계기반 방법이 있다. 시간기반 인력 프로파일과 관련된 대표적인 연구로는 Putnam의 Rayleigh 분포 모델[16]이 있다. 이 모델의 단점을 보완하고자 와이블[12], Gamma[14] 분포 모델이 제안되었다. 이 모델들은 하드웨어 개발 방법에 적합한 모델을 소프트웨어에 적용한 사례로 현실적으로 소프트웨어 개발 과정에서 투입되는 인력 프로파일을 적절히 표현하지 못하는 단점을 갖고 있다. 단계기반 인력 프로파일과 관련된 연구로는 Clark[5]의 COCOMO II 활용 방법, UP[13], Catherine[4]의 폭포수 모델 기반 방법이 있다. 이 방법을 적용시 각 단계에 할당된 총 인력에 대해 다시 단위시간별로 어떤 분포로 인력을 할당할 것인가가 문제로 대두된다.

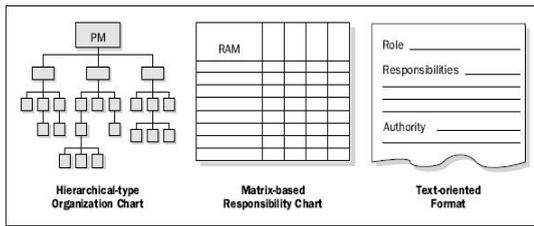
넓은 의미에서 소프트웨어 개발 업무는 일종의 프로젝트이고, 소프트웨어 개발에 참여하는 인력관리는 프로젝트 인적자원 관리와 상통한다고 볼 수 있다[10]. 인적자원 관리를 포함한 프로젝트 관리 전반에 대한 모범적인 실무관행을 정리한 프로젝트관리 지식체계(PMBOK) 지침서가 널리 알려져 있다. 이 지침서는 프로젝트 관리 전문 분야에서 인정받는 표준으로, 이 지침은 확립된 규범, 방법, 프로세스 및 실무관행을 설명하는 공식적인 문서이다[15].

PMBOK에서는 인적자원 관리 프로세스를 <표 1>과 같이 4개의 프로세스와 각 프로세스별 입력, 도구와 기법, 출력에 대한 정의가 되어 있다.

본 논문의 연구 범위는 두 번째 프로세스인 프로젝트 팀 확보 단계에 해당하며, PMBOK에 의하면 [그림 1]과 같은 계층조직도, 매트릭스 기반의 책임할당도, 역할에 대한 정의서를 작성하게 된다.

〈표 1〉 PBMOK의 인적자원관리

Process	Input	Tool & Technique	Output
1. 인적자원 계획서 개발	활동자원 요구사항 기업환경 요인 조직프로세스 자산	조직도 및 직무기술서 네트워크 조직이론	인적자원 계획서
2. 프로젝트 팀 확보	프로젝트 관리계획서 기업환경 요인 조직프로세스 자산	사전 배정 협상 확보 가상팀	프로젝트 팀원 배정 자원역일표 프로젝트 관리계획서 (갱신)
3. 프로젝트 팀 개발	프로젝트 팀원 배정 프로젝트 관리계획서 자원역일표	대인기술 교육 팀 구축 활동 기본규칙 동일장소 배치 인정 및 보상	팀성과 평가치 기업환경요인(갱신)
4. 프로젝트 팀 관리	프로젝트 팀원 배정 프로젝트 관리계획서 팀성과 평가치 성과보고서 조직프로세스 자산	관찰 및 대화 프로젝트 성과평가 갈등관리 대인관계 기술	기업환경요인(갱신) 조직프로세스 자산 (갱신) 변경요청 프로젝트 관리계획서 (갱신)



〔그림 1〕 인적자원 관리

책임할당 매트릭스(RAM, responsibility assignment matrix)는 수행해야할 작업과 팀원 사이의 관계를 보여 준다. 대규모 프로젝트에서는 다양한 수준으로 RAM을 작성할 수 있다. 특히, <표 2>와 같은 RACI(Responsible, Accountable, Consult, Inform)이라는 도표는 RAM에 다양한 수준의 책임을 자세하게 표시할 수 있어서 자주 사용되고 있다.

〈표 2〉 PBMOK의 책임할당 매트릭스(RACI)

RACI	참여 인력				
	김양	이균	박씨	최모	강균
정의	A	R	I	I	I
설계	I	A	R	C	C
개발	I	A	R	C	C
테스트	A	I	I	R	I

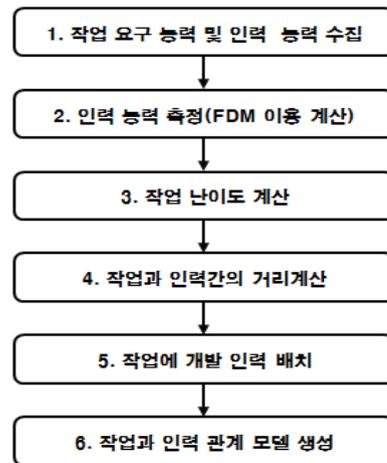
본 논문에서는 소프트웨어 개발 프로젝트에 참여하는

인적 자원에 대한 세부사항이 결정되지 않은 기초 수준의 프로젝트 관리 계획서, 소프트웨어 개발 프로젝트를 완료하기 위해 필요한 능력과 전문성을 고려한 기업 환경 요인을 바탕으로 참여 인력의 업무 배정 방식을 수학적 논리로 결정하기 위한 방안을 제시한다.

### 3. 수학적 업무 배정 방법

#### 3.1 인력 할당 프로세스

소프트웨어 개발업무에서 인력을 할당할 때는 프로젝트 관리론의 인적자원 관리 과정에서 프로젝트 팀 확보 단계에서 사용되는 방법을 적용한다. 본 논문에서는 이 과정에서 프로젝트 관리자나 프로젝트 수주팀의 휴리스틱 방법이 아닌 계량적인 데이터를 근거로 결정하도록 한다. 이를 위해 프로젝트 수행에 필요한 요구 능력과 프로젝트에 참여하게 되는 인력의 능력 수준을 숫자로 계산한다. 또한, 이 값을 바탕으로 개발팀원의 인력 능력, 단위 업무별 작업 난이도, 작업과 인력간의 능력의 차이 값을 계산한다. 이런 값을 이용하여 개발 인력을 배치하고 작업과 인력 관계 모델을 생성한다. 이와 같은 인력 할당 프로세스에 대한 전체 흐름은 [그림 2]와 같다.



〔그림 2〕 인력 할당 프로세스

#### 3.2 작업 요구 능력 및 팀원 능력 수집

소프트웨어 개발 프로젝트에서 해당 업무에 대한 분석이 선행적으로 필요하다. 요구 분석이라는 업무는 소프트웨어 개발 과정에서 가장 어려운 활동이며, 업무 도

메인에 대한 이해 뿐만 아니라 현업에 대한 관찰, 인터뷰, 설문조사 등 굉장히 많은 세부 활동을 수행해야 한다. 기존의 소프트웨어공학에서 다루고 있기에 본 논문에서는 요구 분석을 위한 세부 태스크에 대해서는 언급하지 않는다.

본 논문에서는 소프트웨어 개발 인력 할당을 위한 수학적 접근법에 사용하기 위해 작업 요구 능력과 팀원 능력을 행렬로 표현한다. 개발할 업무를 분해한 결과 필요한 작업이  $m$ 개이고, 각 작업에 필요한 업무 능력의 종류가  $l$  개가 있다면 이를 (수식 1)과 같이 행렬  $R$ 로 표시한다. ( $m$ =작업 수,  $l$ =업무 능력 종류)

$$R = \begin{bmatrix} r_{11} & \dots & r_{1l} \\ \vdots & \ddots & \vdots \\ r_{m1} & \dots & r_{ml} \end{bmatrix} \quad (\text{수식 1})$$

이때 작업에 필요한 능력을 표시하기 위해서는 5점 척도 방법을 이용하여 표시한다. 행렬  $R$ 에서 나타내는 것은 작업  $m$ 에  $k$ 번째 업무를 수행하기 위해 필요한 능력수준에 대한 5점 척도의 값이다. 이때 5점 척도에서 5점은 가장 능력이 좋은 것을 나타내고, 1점은 능력이 가장 나쁨을 나타낸다.

본 논문에서는 개발에 참여할 인력의 능력을 (수식 2)와 같이 행렬  $H$ 로 표시한다. 개발 인력의 능력을 표시하기 위해서는 5점 척도 방법을 이용하여 표시한다. 행렬  $H$ 에서 나타내는 것은 참여 인력이  $n$ 명이고, 그들이 가지고 있는 능력의 종류가  $l$  개 일 때, 개발 인력의 능력을 5점 척도로 나타낸 값이다. ( $n$ =직원 수,  $l$ =업무 능력 종류)

$$H = \begin{bmatrix} h_{11} & \dots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{l1} & \dots & h_{ln} \end{bmatrix} \quad (\text{수식 2})$$

### 3.3 팀원능력 측정

소프트웨어 개발 업무 각각에 가장 적절한 인력을 배치하기 위해서 작업 요구능력 행렬  $R$ 과 개발인력 능력 행렬  $H$ 를 이용하여 우선순위를 결정한다. 우선순위를 계산하기 위해 6-시그마[2] 기법에서 소개하고 있는 기능전개 매트릭스(FDM, Functional Deployment Matrix) 기법을 이용한다. FDM은 입력변수  $X$ 와 최종 고객이 중시하는 출력변수  $Y$ 와의 관계를 이용하여 우선순위를 결정하

는 방법이다. 작업 요구능력 행렬과 개발인력 행렬을 (수식 3)과 같은 FDM을 이용하여  $n$  명의 개발자가  $m$ 개의 작업에 참여할 때 얻어낼 수 있는 기능의 수준은 (수식 4)와 같이 계산할 수 있다.

$$fdm_{ij} = \sum_{k=1}^l (r_{ik} \times h_{kj}) \quad (\text{수식 3})$$

$$FDM = \begin{bmatrix} fdm_{11} & \dots & fdm_{1n} \\ \vdots & \ddots & \vdots \\ fdm_{m1} & \dots & fdm_{mn} \end{bmatrix} \quad (\text{수식 4})$$

개발인력 능력 행렬과 개발인력 FDM 행렬을 이용하여 소프트웨어 개발에 필요한 개발 인력의 능력을 순위로 정한다. 순위를 정하는 방법은 단위 작업에 대해  $n$ 명의  $fdm$  계산값 중에서 값이 가장 높은 것을 1위로 정하고 (수식 5), 만약 같은 경우는 순위를 같게 한다. (수식 6)은 모든 작업에 대해 계산된  $Rank\ FDM$  행렬이다.

$$Rank\ fdm_{ij} = Rank\ of\ fdm_{ij}\ in\ row\ i \quad (\text{수식 5})$$

$$Rank\ FDM = \begin{bmatrix} Rank\ fdm_{11} & \dots & Rank\ fdm_{1n} \\ \vdots & \ddots & \vdots \\ Rank\ fdm_{m1} & \dots & Rank\ fdm_{mn} \end{bmatrix} \quad (\text{수식 6})$$

계산된  $Rank\ FDM$  행렬을 이용하여 각 인력의 평균  $fdm$ (Average  $fdm$ )을 (수식 7)과 같이 계산한다. 개발 인력이 가지고 있는 종합능력은 평균  $fdm$  과 비교하여 평가한다.

$$Average\ fdm_j = \frac{\sum_{i=1}^m Rank\ of\ fdm_{ij}}{m} \quad (\text{수식 7})$$

개발 인력이 가지고 있는 능력의 우선순위를 (수식 8)을 이용해서 계산한다. 개발 능력의 우선순위가 높은 것은 다른 개발 인력에 비해 능력이 높음을 나타낸다.

$$Rank\ of\ HR = [HR\ with\ Top\ Rank\ of\ Average\ fdm, \\ Second\ Rank\ of\ Average\ fdm, \dots, \\ Last\ Rank\ of\ Average\ fdm] \quad (\text{수식 8})$$

### 3.4 작업 난이도 순위 계산하기

개발하고자 하는 소프트웨어 난이도에 따라 적당한 능력의 개발자를 할당하는 일이 필요하며 이를 위해서는 작업 난이도를 계산하는 방법이 고안되어야 한다. 본 논

문에서는 단위 작업 요구 능력의 점수의 합을 작업 난이도로 사용한다. 작업에서 요구하는 능력 요구 점수의 합이 높으면 작업이 어렵다고 볼 수 있다. 작업 난이도 계산 방법은 (수식 9)와 같다.

$$Difficulty = \left[ \sum_{j=1}^l r_{1j} \cdots \sum_{j=1}^l r_{mj} \right] \quad (\text{수식 } 9)$$

### 3.5 개발 인력 능력과 작업요구 능력간의 거리 계산

소프트웨어 개발 개별 단위 작업에 요구되는 능력과 개발 인력이 가지고 있는 능력 간에 차이 값을 계산을 할 필요가 있다. 단위 개발 업무에 너무 고급 인력을 배치하거나, 너무 능력이 떨어지는 인력을 배치시키는 일을 막음으로서 효율적이며 최적의 인력 할당이 가능하다. 본 논문에서는 이 차이 값의 계산을 (수식 10)과 같이 거리라고 표현한다. 거리계산 행렬은 (수식 11)과 같다.

$$distance_{ij} = \sum_{k=1}^l (h_{ki} - r_{jk}) \quad (\text{수식 } 10)$$

$$Distance = \begin{bmatrix} distance_{11} & \cdots & distance_{1m} \\ \vdots & \ddots & \vdots \\ distance_{n1} & \cdots & distance_{nm} \end{bmatrix} \quad (\text{수식 } 11)$$

### 3.6 단위 작업에 개발인력 배치하기

본 논문에서는 단위 작업에 개발 인력을 배치하는 방법으로 (수식 8)의 인력 종합능력 순위와 (수식 9)의 작업 난이도 순위를 사용한다.

우선 계산된 작업 난이도 순위를 이용하여 가장 쉬운 단위 작업을 식별하고, 이 작업에 인력 종합능력 순위 중 가장 약한 개발 인력을 배치한다. 능력이 가장 좋은 개발 인력은 마지막 가장 어려운 단위 작업에 배치한다. 나머지 작업도 작업난이도와 인력 종합순위를 이용하여 차례대로 배치한다.

단위 작업에 개발 인력을 배치한 초기 할당 결과의 타당성을 검증하는 것도 좋은 방안이다. 검증할 때는 3.4절에서 계산할 수 있는 거리계산 값을 이용한다. 거리 계산 결과가 0이상이면 단위 작업에 할당된 개발인력이 해당 단위 업무를 처리할 수 있는 능력이 된다는 것이며, 만약 0미만이면 단위 작업에 할당된 개발 인력이 해당 작업을 혼자 완수할 수 없고, 다른 개발 인력의 도움이 필요함을 나타낸다. 즉 능력 있는 인력과의 유기적인 협력관계를

통하여 작업을 할 필요가 있음을 나타낸다. 인력 능력과 단위 작업에 필요한 능력간에 차이가 많이 나면 인력을 교체하거나 다른 인력이 대신 해주어야 된다.

어떤 프로젝트는 개발에 참여하는 인력 수( $n$ )와 단위 작업개수( $m$ )가 다를 수 있다. 이런 경우, 다음과 같은 인력 배치 방법을 사용한다.

가) 인력 수  $n$ 가 단위 업무 갯수  $m$ 보다 더 많은 경우  
While  $n > m$  do

tasks are arranged into the order of difficulty  
(difficult->easy)

assign the corresponding human into the task  
if  $Distance_{ii} < 0$

then the task should be matched  
the remaining human

else the matching is correct

the remaining human do auxiliary work

나) 인력 수  $n$ 가 단위 업무 갯수  $m$ 보다 더 적은 경우  
While  $n < m$  do

tasks are arranged into the order of difficulty  
(difficult->easy)

arrange the corresponding human

if  $Distance_{ii} < 0$

then the task should be matched  
the remaining human

else the matching is correct

the remaining task could be done by human with  
sufficient capacity

### 3.7 인력 배치 적절성 검증하기 위해 개발인력 관계모델 작성하기

협력적인 개발팀 활동에 있어서 의사소통은 중요한 부분임에 틀림이 없다. 의사소통은 그것이 이루어지는 상황과 대상에 따라서 그 의미가 다르게 해석될 수 있는데, 그 의미가 명확하게 전달되지 못하는 경우가 매우 많다.

유기적인 협력관계를 평가하기 위해 갈등 수준에 따른 팀 성과 관계를 활용한다. 팀에서 인력들이 서로 교류가 적거나 많으면 갈등수준이 낮거나 높게 나타난다. <표 3>과 같이 갈등수준이 낮거나 높으면 팀의 성과도

낮게 나타난다[9]. 팀에서 적당히 커뮤니케이션이 있으면 갈등 수준이 보통이 되고 팀의 효율성과 성과도 더 높게 나타난다. 개발 인력간의 유기적인 협력관계를 평가하기 위해 앞 절에서 할당된 인력과 단위 작업간의 관계모델을 만들고 갈등수준을 계산한다. 제시한 인력배치 방안이 팀워크의 효율성을 증진시키는 방안이라는 것을 증명하기 위해 개발인력 관계모델을 이용하여 갈등수준을 계산한 후, 제시된 방법이 갈등수준의 보통임을 판별한다.

〈표 3〉 갈등과 팀의 성능 수준 관계

갈등수준	낮음	보통	높음
팀 성과	낮음	높음	낮음

본 논문에서는 소프트웨어 개발 인력간의 관계망을 분석한다. 관계망은 인력은 노드(Node)로, 인력간의 관계 형성은 간선(Edge)로 표현한다.

개발 인력 관계 모델 형성은 다음의 과정을 거친다. 우선 관계 모델의 중심노드인 프로젝트 리더를 선정해야 한다. 프로젝트 리더를 선정하는 방법은 3.3절에서 제시한 FDM을 이용하여 개발인력의 능력치 중 가장 높은 능력 값을 가진 개발 인력을 선정한다. 선정된 프로젝트 리더는 관계 모델에서 개발 인력들 간의 유기적 협력관계를 형성하는 중요한 역할을 한다. 개발인력 관계 모델에서 선정된 프로젝트 리더는 중심노드로 다른 개발 인력과 모두 간선으로 연결된다.

개발 인력 관계모델 생성을 위한 두 번째 단계는, 중심노드를 제외한 개발 인력들 간의 연결 관계를 설정한다. 관계를 설정하기 위해 3.5절에서 제시한 작업요구 능력과 개발 인력 능력의 거리 테이블을 이용한다. 인력간의 간선을 만들기 위한 기준 값은 (수식 12)과 같이 가장 능력이 있는 인력(j)의 작업별 거리를 모두 합한 값을 작업의 수로 나눈 후 1의 값을 뺀다. 인력들 간에 간선은 작업-인력 테이블을 이용하여 각 요소의 값이 기준 값 이상인 경우에 설정한다.

$$\text{기준값} = \frac{\sum_{k=1}^l \text{distance}_{kj}}{m} - 1 \quad (\text{수식 } 12)$$

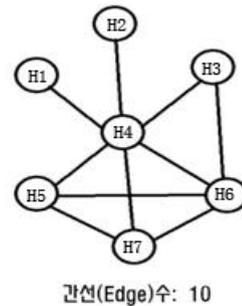
개발 인력관계 모델이 가질 수 있는 최대 간선의 수는  $\frac{n \times (n-1)}{2}$  개이다. 간선의 수를 이용하여 갈등수준을 계

산하기 위해 (수식 13)식을 이용한다.

$$\left[ \frac{\frac{n \times (n-1)}{2}}{2} \right] \pm \left[ \frac{n}{2} \right] \quad (\text{수식 } 13)$$

간선의 수가 (수식 13)의 범위에 해당할 경우는 갈등수준이 보통으로 판단하고, (수식 13)의 범위 보다 크거나 작은 경우에는 갈등 수준이 높다고 판정한다.

제안된 모델의 인력 관계모델을 설명하기 위해 [그림 3]과 같은 팀 구조를 가정할 때, 간선의 수를 계산하면 갈등수준이(7~14범위 내) 보통으로 나타나는 것을 알 수 있다. 그러므로 본 논문에서 제안한 인력 관계 모델을 이용하여 인력을 배치하면 소프트웨어 개발시 유기적인 협력관계가 이루어지게 되어 개발효율과 생산성을 향상시킬 수 있다.



[그림 3] 인력관계 모델 예

## 4. 적용 사례

### 4.1 개요

소프트웨어 개발 인력 배치를 위한 수학적 업무 배정 방법의 적용 가능성을 검토하기 위하여 적용 사례를 소개한다. 적용 프로젝트는 무역회사의 ERP 시스템을 개발하는 범위로, 크게 7개의 단위 업무를 전산화 하기 위해 5개의 소프트웨어 개발 능력이 필요하고, 7명의 개발자를 배치하고자 한다.

### 4.2 작업 요구 능력 및 팀원 능력 수집

개발하고자 하는 프로젝트는 7개의 작업으로 분해되었다. 분해된 작업을 R1~R7로 표시 하며, 각각의 작업

은 아래와 같다.

- R1 : 프로젝트 리더
- R2 : 프로젝트 조직 및 관리
- R3 : 해외 무역 업무 관리 시스템
- R4 : 재무 관리 시스템
- R5 : 보관 및 운송 관리 시스템
- R6 : 인적 자원 관리 시스템
- R7 : 회사 오피스 작업

소프트웨어 개발 업무 수행을 위해 필요한 능력이 5가지가 있다. 단위 작업에 필요한 능력을 (a1,a2,a3,a4,a5)로 표시하며 세부적인 능력은 아래와 같다.

- a1 : 조직 및 관리
- a2 : 시스템 분석
- a3 : 프로그래밍 기술
- a4 : 의사소통
- a5 : 시스템 테스트

단위 작업에 필요로 하는 각각의 능력을 5점 척도(1: 나쁨 ~ 5: 아주 좋음)를 이용하여 값을 표시한다. 프로젝트의 7개 단위 작업에 각각의 필요한 능력 값은 아래 <표 4>와 같다.

<표 4> 단위 작업에 필요한 능력

작업 \ 능력	a1	a2	a3	a4	a5
R1	5	4	3	4	4
R2	4	5	4	3	2
R3	2	4	3	2	2
R4	2	3	4	2	2
R5	1	3	4	3	3
R6	1	3	3	3	2
R7	3	3	3	2	2

프로젝트 개발에 참여하는 인원은 총7명으로, 각각을 H1 ~ H7로 표시한다. 개발인력이 가지고 있는 각각의 능력을 5점 척도(1: 나쁨 ~ 5: 아주 좋음)를 이용하여 값을 표시한다. 개발 인력 7인이 가지고 있는 능력 값은 아래 <표 5>와 같다.

<표 5> 개발인력이 가진 능력

인력 \ 능력	H1	H2	H3	H4	H5	H6	H7
a1	3	1	2	2	3	4	5
a2	3	3	3	4	3	5	4
a3	4	4	3	4	4	5	3
a4	3	3	4	3	4	4	4
a5	2	3	3	4	3	3	4

### 4.3 인력 능력 순위 계산

인력의 능력을 순위를 계산하기 위해 FDM계산하는데 이때 (수식 3)을 이용한다. 한 예로 작업 R1에서 인력 H1, H2의 FDM계산은 아래와 같다.

$$f_{dm_{ij}} = \sum_{k=1}^l (r_{ik} \times h_{kj})$$

$$FDM\ R1H1=(3 \times 5 + 3 \times 4 + 4 \times 3 + 3 \times 4 + 2 \times 4) = 59$$

$$FDM\ R1H2=(1 \times 5 + 3 \times 4 + 4 \times 3 + 3 \times 4 + 3 \times 4) = 53$$

위의 과정을 이용하여 작업 R1에서 FDM을 계산하면 RF1=( 59 53 59 66 67 83 73) 로 계산할 수 있다. 모든 작업의 FDM계산 결과는 아래 <표 6>과 같다.

<표 6> 작업의 FDM 계산 결과

	H1	H2	H3	H4	H5	H6	H7
RF1	59	53	59	66	67	83	73
RF2	56	50	53	61	61	79	66
RF3	44	42	42	50	48	62	50
RF4	41	39	39	46	45	57	47
RF5	46	45	46	53	52	64	53
RF6	37	37	38	43	42	52	41
RF7	40	36	38	44	44	56	48

모든 작업에 대한 FDM 계산 결과를 (수식 5)를 이용하여 작업에 대한 인력의 순위를 정한다. 모든 작업에 대한 Rank of HR (각 작업에서 인력의 우선순위)결과는 <표 7>과 같다.

<표 7> 작업에서 인력의 우선순위

	H1	H2	H3	H4	H5	H6	H7
RHR1	5	7	5	4	3	1	2
RHR2	5	7	6	3	3	1	2
RHR3	5	6	6	2	4	1	2
RHR4	5	6	6	3	4	1	2
RHR5	5	7	5	2	4	1	2
RHR6	6	6	5	2	3	1	4
RHR7	5	7	6	3	3	1	2

<표 7>의 작업에서 인력의 우선순위를 이용하여 인력의 종합능력을 계산한다. 계산하는 방법은 (수식 7)을 이용한다. 각 인력의 Average FDM값은 <표 8>과 같다. 이때 Average FDM값이 1에 가까울수록 능력 좋음을 나타낸다. 인력 중 능력이 가장 출중한 인력은 H6임을 알 수 있다.

$$Average\ fdm_j = \frac{5+5+5+5+5+6+5}{7} = 5.1$$

<표 8> 모든 인력의 FDM우선순위 평균값

	H1	H2	H3	H4	H5	H6	H7
Average FDM	5.1	6.6	5.6	2.7	3.4	1	2.3

그래서 인력 능력 순위는 H2<H3<H1<H5<H4<H7<H6 이다.

#### 4.4 작업 난이도 순위 계산

(수식 9)를 이용하여 각 작업의 난이도를 계산 한다. 계산된 작업난이도의 결과는 <표 9>와 같다. 작업 난이도의 순위를 표시하면 R6 < R4 = R7 < R3 < R5 < R2 < R1과 같다. 예를 들면, R1의 작업 난이도는 (5+4+3+4+4)=20 이다.

<표 9> 각 작업의 난이도 점수

	R1	R2	R3	R4	R5	R6	R7
난이도	20	18	14	13	15	12	13

#### 4.5 개발 인력 능력과 작업요구 능력간의 거리 계산

(수식 10)을 이용하여 인력(H)과 작업(R)간의 거리를 계산한다.

$$distance_{ij} = \sum_{k=1}^5 (h_{ki} - r_{jk})$$

$$distance_{ij} = (3-5)+(3-4)+(4-3)+(3-4)+(2-4)=-5$$

작업 R1과 모든 인력의 거리 RD1= [-5, -6, -5, -3, -3, 1, -2] 이다. 모든 작업과 모든 인력의 거리는 <표 10>과 같다.

<표 10> 작업-인력 거리 결과

	H1	H2	H3	H4	H5	H6	H7
RD1	-5	-6	-5	-3	-3	1	-2
RD2	-3	-4	-3	-1	-1	3	0
RD3	1	0	1	3	3	7	4
RD4	2	1	2	4	4	8	5
RD5	0	-1	0	2	2	6	3
RD6	3	2	3	5	5	9	6
RD7	2	1	2	4	4	8	5

#### 4.6 작업에 개발인력 배치하기

작업에 인력을 배치하기 위해서 먼저 쉬운 작업에 낮은 능력을 가진 인력을 배치한다. <표 8>과 <표 9>을 이용하여 배치를 하면 작업 R6에 인력 H2가 배치된다. 마찬가지로 작업 순위가 그 다음 순위인 R4과 R7과 같이 공동일 경우 능력이 낮은 H3과 H1을 우선 선택한 후, 거리 계산을 한다. 계산된 결과를 이용하여 작업에 배치를 하면 작업 R4에 인력 H3을 작업 R7에 H1을 배치한다. 이런 과정을 거쳐 만들어진 인력-작업 배치결과는 <표 11>과 같다.

<표 11> 단위 작업에 개발인력 할당 결과

인력능력순위 (약-->강)	H2	H3	H1	H5	H4	H7	H6
작업난이도순위	R6	R4	R7	R3	R5	R2	R1

H1-->R7:회사 오피스 작업

H2-->R6:인적 자원 관리 시스템 작업

H3-->R4:재무 관리 시스템 작업

H4-->R5:보관 및 운송 관리 시스템 작업

H5-->R3:해외 무역 업무 관리 시스템 작업

H6-->R1:리더 결정 작업

H7-->R2:프로젝트 조직 및 관리 작업

#### 4.7 팀 조직 협력 관계 모델 만들기

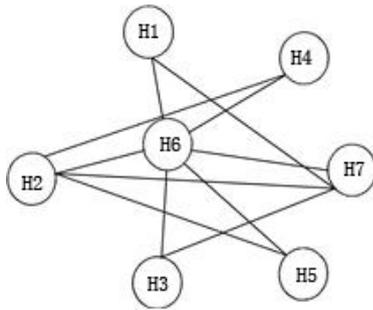
4.2절에서 계산된 개발인력의 능력치 중 가장 높은 능력 값을 가진 개발 인력은 H6이다. 선정된 프로젝트 리더는 중심노드로 다른 개발 인력과 모두 간선으로 연결된다.

관계를 설정하기 위해 <표 12> 인력-작업 거리 테이블을 이용한다. 인력간의 간선을 만들기 위한 기준 값은  $\left\lceil \frac{1+3+7+8+6+9+8}{7} \right\rceil - 1 = 5$  이다. 인력들 간에 간선은 작업-인력 테이블을 이용하여 각 요소의 값이 기준

값 이상인 경우는 인력과 작업 할당인력은 연결한다. 해당되는 인력의 연결 모델은 아래 [그림 4]와 같다.

〈표 12〉 인력-작업(할당인력) 거리 테이블

	H1	H2	H3	H4	H5	H6 (최강)	H7
R1/V6	-5	-6	-5	-3	-3	1	-2
R2/V7	-3	-4	-3	-1	-1	3	0
R3/V5	1	0	1	3	3	7	4
R4/V3	2	1	2	4	4	8	5
R5/V4	0	-1	0	2	2	6	3
R6/V2	3	2	3	5	5	9	6
R7/V1	2	1	2	4	4	8	5



[그림 4] 연결된 인력 관계 모델

본 논문에서는 갈등수준 계산하기 위해 개발 인력 관계 모델의 간선의 수를 이용한다. 개발 인력관계 모델이 가질 수 있는 최대 간선의 수는  $\frac{7 \times (7-1)}{2} = 21$ 개가 된다. 간선의 수를 이용하여 갈등수준을 계산하는 식은 다음과 같다.

$$\frac{7 \times (7-1)}{2} \pm \left\lceil \frac{7}{2} \right\rceil$$

본 적용 예제에서 간선의 수가 7~15의 범위에 해당할 경우는 갈등 수준이 보통으로 판단한다. 위 [그림 8]에서 간선수가 11 이기 때문에 갈등수준이 보통 범위에 있고 효율성 높고 팀 성과가 일어난다고 볼 수 있다.

## 5. 결론

기존에는 직관적이며 휴리스틱한 방법으로 소프트웨

어 개발 프로젝트의 인력을 할당하였으나, 본 논문에서는 계량적인 데이터를 근거로 결정할 수 있는 새로운 접근법 제안하였다. 이 접근법은 크게 6개의 프로세스로 구성되어 있으며, 각 단계별로 매트릭스 기반으로 데이터를 계산할 수 있게 하였다. 개발자 인원 수와 단위 작업 숫자가 같은 경우의 프로젝트에 적용하는 사례를 통해 이 접근법을 활용할 수 있는 방법을 소개하였다.

본 논문의 연구 결과는 회사의 포트폴리오의 전략적 사업 목적 달성을 위해 소프트웨어 개발 인력을 효율적으로 배치하는 목적으로 사용될 수 있다. 소프트웨어 개발은 많은 인원이 오랜 시간동안 공동의 작업을 해야 하는 경우가 일반적이므로, 업무의 진척 정도와 시간의 진행에 따라 인력의 변동이 발생할 때 어떻게 대처해야 하는가, 실제 개발 현장에 적용했을 때의 효과에 대한 추가적인 연구가 필요하다.

## 참고 문헌

- [1] 김양숙 (2007). 소규모 IT Project의 PMBOK 기반의 적절한 Human Resource 방안 제시, 아주대학교 석사학위 논문
- [2] 류한석 (2005). 프로젝트 관리에서 문제점 우선순위 결정 기법, 정보과학회 가을 학술발표집, 32(2), 301-303.
- [3] 박석규 (2007). 소프트웨어 개발 인력 배분 모델, 한국인터넷 정보학회지, 8(2), 23-31.
- [4] Cockburn, A. and Highsmith, A. (2001). Agile Software Development: The People Factor, IEEE Computer, 34(11), 131-133.
- [5] Catherine, M. B., "TSM 5316 - Project Management," Computer Information Systems, College of Business, Florida Gulf Coast University, 2002.
- [6] Clark, B., (1998). COCOMO II Model, CSE Research Associate 15th COCOMO/SCM Forum.
- [7] Cohn, M. (2009). Succeeding with Agile: Software Development Using Scrum, Addison-Wesley.
- [8] Dyba, T. and Dingsøyr, T. (2009), Empirical studies of agile software development: A systematic review, Information and Software Technology, 50(5), 833-859.

[9] Hao, D. and Liu, M. (2005), The research group's management based on the structure of its cohesion system, *Science and Technology Management Research*, 25(11), 87-89.

[10] Huemann, M., Keegan A. and Turner, J. R. (2007), Human resource management in the project-oriented company: A review, *International Journal of Project Management*, 25, 315-323.

[11] Kerzner, H. (2012). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, John Wiley & Sons.

[12] Knafl, G. J. and Gonzales, C. (1998). An Evaluation of Software Cost Models, <http://facweb.cs.depaul.edu/research/techreports/TR98-005.doc>, DePaul University.

[13] Kruchten, P. (2002). "Planning an Iterative Project," *The Rational Edge e-zine for the Rational Community*, <http://www.ibm.com/developerworks/rational/library/2831.html>.

[14] Pillai, K. and Sukumaran Nair, V. S. (1997). A Model for Software Development Effort and Cost Estimation, *IEEE Trans. on Software Eng.*, 23(8), 485-497.

[15] PMBOK (2008), *Guide to the Project Management Body of Knowledge 4th*, Project Management Institute.

[16] Putnam, L. H. (1978). A General Empirical Solution to the Macro Software Sizing and Estimating Problem, *IEEE Trans. on Software Eng.*, 4(4), 345-361.

[17] Salo, O., Abrahamsson, P. (2008), Agile Methods in European Embedded Development Organizations: a survey study of Extreme Programming and Scrum, *IET Software*, 2(1), 58-64.

[18] Schwaber, C. (2009). *Enterprise Agile Adoption in 2007*, Forrester Research.

[19] Sutherland, J., Viktorov, A., Blount, J. and Puntikov, N. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams, *Proceedings of the 40th HICSS*, 1-10.

### 진 상



- 2011년 2월 : 우송대학교 게임멀티미디어학과(공학사)
- 2011년 3월~현재 : 전남대학교 대학원 전자컴퓨터공학 석사과정
- 관심분야 : 소프트웨어공학, IT 서비스, 스마트컴퓨팅, 전자상거래 등
- E-Mail : cxbsr2006@gmail.com

### 이 상 준



- 1991년 2월 : 전남대학교 전산통계학과(이학사)
- 1993년 2월 : 전남대학교 전산통계학과(이학석사)
- 1999년 8월 : 전남대학교 전산통계학과(이학박사)
- 2007년 2월~현재 : 전남대학교 경영학과 부교수

· 관심분야 : 경영정보시스템, 스마트컴퓨팅, 소프트웨어공학

· E-Mail : s-lee@chonnam.ac.kr

### 서 성 채



- 1994년 2월 : 전남대학교 전산학과(이학사)
- 1997년 2월 : 전남대학교 전산통계학과(이학석사)
- 2006년 2월 : 전남대학교 전산통계학과(이학박사)
- 2007년 3월~현재 : 전남대학교 유비쿼터스 정보가전사업단 박사후연구원

· 관심분야 : 소프트웨어공학, 테스트, 품질, 객체지향

· E-Mail : scseo@chonnam.ac.kr

### 김 병 기



- 1978년 2월 : 전남대학교 수학과(이학사)
- 1980년 2월 : 전남대학교 수학과(이학석사)
- 2000년 2월 : 전북대학교 수학과(이학박사)
- 1981년 2월~현재 : 전남대학교 전자컴퓨터공학부 교수

· 관심분야 : 소프트웨어공학, 프로젝트 관리

· E-Mail : bgkime@chonnam.ac.kr