

Research And Improve Agile Development Methods Based On Teamwork

Chen Xiang[†] · Byungki Kim^{**}

ABSTRACT

Agile software development methodology is a lightweight software development method, plays an important role in the software development team. In this paper, we propose programs that in when agile development co-operation, team management, team collaboration and project management tasks can be high efficiency and allocation of human resources. The proposed model, the use of 6-Sigma priority assignment, established a mathematical model of the operation and distribution of human resources is the most reasonable people go to work. Cooperation makes team-based agile development can improve efficiency. This method improve staff efficiency by teamwork to improve the productivity of software development.

Keywords : Agile, Teamwork, Prioritization

팀워크 바탕으로 애자일 개발방법의 개선연구

Chen Xiang[†] · 김 병 기^{**}

요 약

애자일 소프트웨어 개발 방법은 경량 소프트웨어 개발 방법으로 소프트웨어 개발에서 팀워크가 중요한 역할을 한다. 본 논문에서는 애자일 개발에서 공동 작업을 할 때 팀워크, 팀 관리, 프로젝트 관리를 효율적으로 수행할 수 있는 작업과 인적자원을 분배하는 방안을 제시한다. 제안 모델은 작업과 인적자원 분배를 위한 수학적 모델을 제시하고 6-시그마의 우선순위를 적용해서 작업에 가장 합당한 인력을 할당한다. 그리고 개발인력 관계모델을 이용하여 작업에 할당된 인력의 적절성을 검증한다. 제안된 방법은 인원배치의 효율성을 통한 팀워크를 개선함으로써 SW 개발 생산성을 높였다.

키워드 : 애자일, 팀워크, 우선순위

1. 서 론

애자일 방법론[1]은 짧은 주기의 반복적이고 지속적인 제품 출시를 통해 시장의 반응을 빠르게 적용하며, 전문성이 높은 소규모의 팀을 구성하여 효과적인 대화와 협력으로 생산성을 극대화한다.

애자일 개발 방법의 원칙은 개발자 중심으로 고객과 개발자들이 빈번한 대화를 통해 대화를 함으로써 신속하고, 연속적으로 개발하는 방법이다. 애자일 개발 방법에서 개발에 참여하는 개발자가 많은 경우 팀워크의 중요성이 증가한다.

애자일 개발방법에서 팀워크 유지를 통한 생산성과 품질을 개선하기 위해서는 팀워크 유지가 필수적이다. 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[2]에서는 작업에 가장 적합한 능력을 가진 인적자원을 할당하는 것은 개발의 효율성을 증진한다고 언급하고 있지만, 작업과 인적자원 할당에 대한 논의는 거의 없는 실정이다.

본 연구에서는 애자일 개발방법에서 개발되는 프로젝트에서 작업과 개발자의 분배 문제점을 극복하기 위해 팀워크 개념을 추가한 방법을 제안한다. 제안된 모델은 작업과 인적자원 분배를 위해 수학적 모델을 만들고 6-시그마의 FDM우선순위 계산[3]과 거리계산 등을 적용해서 작업에 가장 합당한 사람을 할당한다. 그리고 개발인력 관계모델을 이용하여 작업에 할당된 인력의 배치의 적절성을 검증한다. 이를 통해 인원배치의 효율성을 통한 팀워크를 개선함으로써 SW개발 생산성을 높였다.

[†] 준 회원: 전남대학교 전자컴퓨터공학과 대학원생

^{**} 종신회원: 전남대학교 전자컴퓨터공학과 교수

논문접수: 2012년 12월 24일

심사완료: 2013년 1월 28일

* Corresponding Author: Byungki Kim(bgkim@jnu.ac.kr)

2. 관련연구

2.1 애자일 개발방법과 문제점

1) 애자일 개발 방법개요

애자일 소프트웨어 개발 방법은 전통적인 소프트웨어 개발 방법에 비해 경량적인 개발 방법이다. 변화무쌍한 시장과 고객의 요구 변화에 유연하게 대응함으로써 고객의 만족과 비즈니스의 성공을 이끌어내는 데에 그 핵심이 있다. S/W 프로젝트 개발은 새로운 요구 사항은고효율, 고품질, 저렴한 개발 및 생산이 있다. 좋은 S/W 개발 방법은 사람, 기술 및 프로세스는 좋은 조화를 이루고 있어야하고 끊임없이 변화하는 시장의 요구사항 및 고품질을 만족 해야 한다.

애자일 방법을 만든 Martin Fowler는 애자일 개발방법의 핵심개념을 적응과 사람중심이라고 정의하였다[4]. 이 개념을 기반으로 XP[5], FDD[6], 스크럼, 크리스탈 등 많은 새로운 소프트웨어 개발 방법이 제시되었다. 이 방법들은 고품질 소프트웨어의 신속한 전달에 치중하여 고객만족을 보장하고자 하였다. 특히 XP에서는 인간의 요소가 강조되었다.

2)애자일 개발 방법의 문제점

요즘에 프로젝트 규모가 점점 커짐으로 인해, 프로젝트에 참가하는 인원도 점점 증가하고 있다. 애자일 개발 방법의 원칙은 빠르게 인간중심으로 고객과 개발자들이 자주 대화하고 계속 개발하는 것이다. 그러나 프로젝트 규모의 커짐으로 인해 개발자간의 대화 채널이 많아지면서 개발의 신속성을 유지하기 힘들게 된다. 또한 개발의 기민성 유지가 힘들어 개발 시간과 효율성이 저하되는 경향이 발생한다.

2.2 팀워크와 수학적 모델

1) 소프트웨어 개발에서 팀워크의 중요성

컴퓨터 기술의 발달로 컴퓨터 응용 범위는 많이 확대되었다. 과거 혼자서 개발하는 소프트웨어 개발방식에서 벗어나 현재는 지금은 프로젝트 개발자가 수십 명 또는 수백 명으로 구성되어 프로젝트를 진행하고 있다. 프로젝트에 참여하는 사람이 많아지면 역할이 필요하고, 역할 간의 관리가 필요하다. 애자일 개발 방법을 적용한 개발에서는 개발자들은 프로젝트를 완료하기 위해 함께 일하기 좋은 관리환경을 갖추어야 한다. 그러나 활용되고 있는 애자일 개발방법은 인간 요소에 초점을 맞추는 있으니 팀의 역할에 대한 관심과 연구는 미진한 편이다. 이를 개선하기 위해서는 애자일 소프트웨어 개발 방법에 팀워크의 요소를 추가할 필요가 있다. 즉 애자일 방법이 추구하는 인간 요소에 집중뿐만 아니라 팀 협력개선을 통해 개발의 효율성을 증진할 필요가 있다. 효율적인 팀 공동 작업은 개별 개발자의 역할보다 크며, 대규모 프로젝트일수록 반드시 효율적인 팀워크 구축이 필요하다.

팀은 여러 기능과 지식을 가진 사람들이 모여서 작은 그룹을 구성 하고 공통의 목표를 가지고 할 일을 구분해서 서

로 일 하는 것이다[7]. 요즘 프로젝트를 정해진 기간 내에 개발해야하는 데 이때 대부분 팀 협력과 교류를 통하여 개발하고 있다. 팀 간의 교류방식은 구두, 서면, 회의, 협의 방식이 있다. Table 1은 팀의 구성원간의 교류 방식의 특성을 표시한 것이다.

Table 1. The characteristics of the communication team

Mode	Merit	Disadvantages	Adaptability
Verbal	Fast	Unclear	General Questions
Written	Slightly faster	One direction	Unilateral exchange
Meeting	More information	Unclear	Multi-sectoral coordination
Agreement	Official	Long cycle	Multi-company regulation

2)팀의 상호 의존성

프로젝트를 진행할 때 팀 안에는 상호 의존성이 존재하게 된다[8]. 상호 의존성이란 팀 구성원들이 프로젝트를 진행하는 과정에서 발생하는 협력과 상호 작용의 정도를 말한다. Marcus는 상호 의존성은 개별 컬렉션이나 작은 그룹에서 다른 팀 대비의 중요한 특징이라고 지적했다.

Ren[9]의 논문에서는 팀 상호 의존성을 네 개 측면으로 파악하였다.

- a) 팀 작업의 투자; 예를 들어 기술, 자원, 기술의 배포
- b) 팀 멤버의 작업 수행과정
- c) 팀 멤버의 목적과 목표 달성 방식
- d) 팀 멤버의 의견과 임금의 지불 방식

또한 팀 상호 의존성에 대한 네 개 측면을 토대로 상호 의존성의 범위를 크게 작업 상호 의존성과 산출 상호 의존성의 2가지 주요 범주로 나누었다. 작업 상호 의존성은 팀 작업 실행과정을 통하여 결정되고, 산출 상호 의존성은 운영 결과에 의해 결정된다. 이 두 개념은 경험에서 차이 있지만 둘 다 독립적으로 팀 내에 존재할 수 있다.

팀은 고급 조직형태가 상당한 이점을 가지고 있지만 팀에서 어떤 멤버가 자주 커뮤니케이션을 하지 않고, 협력도 하지 않으면 팀원 간의 지식 공유 및 상호 협력이 이루어지지 않는 경우도 발생한다. 이는 프로젝트 조직 운영의 효율성에 영향을 미치고 있다.

Wang[10]은 팀 협업을 강화하기 위한 방법으로 다음과 같이 주장하고 있다.

- a) 동등한 기회를 주고
- b) 공통의 비전을 수립하기
- c) 필요한 교육을 수행
- d) 팀 멤버들에게 위기의식을 강화하기
- e) 필요한 규칙을 개발하기

3)팀워크 관계 모델

팀워크의 이론을 검증하기 위해 Fig. 1과 같은 팀원 관계

모델을 이용한다. 관계 모델은 팀 구성원을 노드를 하고, 구성원간의 관계가 좋을 때는 이를 실선으로 연결하여 그럴 수 있다. Fig. 1은 5명의 구성원간의 관계 모델을 나타내고 있다. Fig. 1의 a는 다일 교류방식이기 때문에 인력간의 협력 관계가 좋지 않는 경우이고 Fig. 1의 b는 커뮤니케이션 방식으로 서로 교류 많고 협력 관계가 좋을 경우임을 알 수 있다.

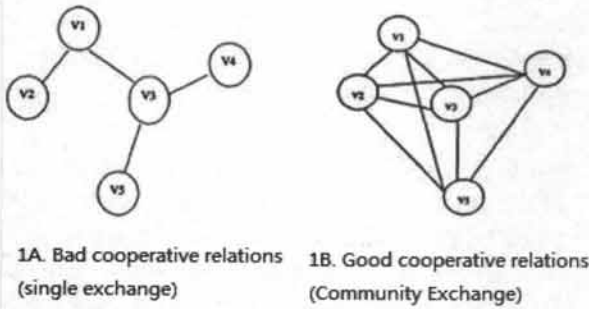


Fig. 1. Relationship Mode by development team of five workforce

관계 모델을 이용하여 Hao[11]은 팀의 교류와 갈등 수준도 낮으면 팀 성과도 낮고, 교류와 갈등 수준이 높으면 팀의 성과 또한 낮아진다고 하였다. 가장 좋은 형태는 팀 교류와 갈등수준이 보통이 되었을 때 가장 좋은 성과가 나타난다고 하였다.

3. 협력기반 팀워크 개선 방안

본 연구에서는 애자일 S/W개발 방법을 이용하여 S/W개발 효율성을 재고할 수 있는 팀원의 유기적 협력관계 모델을 제시 한다. 개괄적인 순서는 Fig. 2와 같다.

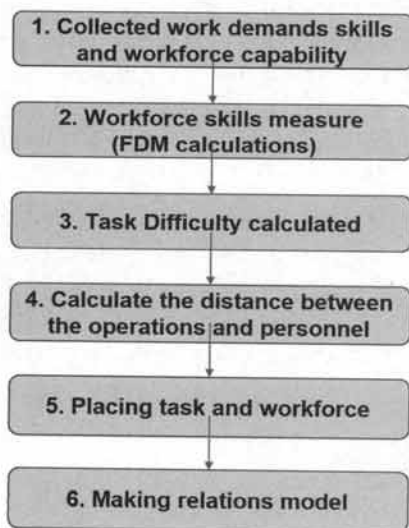


Fig. 2. Method of based on cooperation to improve teamwork

3.1 작업 요구 능력 및 팀원 능력 수집

팀워크를 이용한 애자일 개발 방법을 이용하여 S/W를 개발할 때 유기적인 협력관계를 이용하여 효율적인 개발을 하고자 한다. 이를 위한 첫 단계는 개발 S/W의 작업을 분해해야한다. 그리고 분해된 작업에 필요한 인력 배치가 필수적이다.

개발할 S/W를 분해한 결과 필요한 작업이 n개이고, 각 작업에 필요한 능력이 k개이라면 이를 행렬 R로 Equation(1)과 같이 표시한다. 이 때 작업에 필요한 능력을 표시하기 위해서는 5점 척도 방법을 이용하여 표시한다. 행렬 R에서 나타내는 것은 작업 n에 k번째 능력에 대한 5점 척도의 값이다. 이 때 5점 척도에서 5점은 가장 능력이 좋은 것을 나타내고, 1점은 능력이 가장 나쁨을 나타낸다.

개발에 참여한 인력의 능력을 행렬 V로 Equation(2)와 같이 표시한다. 개발 인력의 능력을 표시하기 위해서는 5점 척도 방법을 이용하여 표시한다. 행렬 V에서 나타내는 것은 작업 k에 대한 m번째 개발 인력의 능력으로써 5점 척도의 값이다.

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1k} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nk} \end{bmatrix} \quad (1)$$

$$V = \begin{bmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{km} \end{bmatrix} \quad (2)$$

3.2 팀원능력 측정(FDM 계산)

작업에 가장 적절한 인력을 배치기 위해서 작업행렬 R과 개발인력 행렬 V를 이용하여 우선순위를 결정한다. 우선순위를 계산하기 위해 6-시그마[2] 기법에서 많이 사용하고 있는 FDM(X-Y Matrix) 기법을 이용한다. FDM은 입력변수 X와 최종 고객이 중시하는 출력변수 Y와의 관계를 이용하여 우선순위를 결정하는 방법이다.

작업행렬과 개발인력 행렬을 FDM으로 계산하는 방법은 아래와 같다. Equation(3)을 이용해서 인력과 개발인력 FDM 행렬을 만들면 Equation(4)와 같다.

$$FDM = \sum_{k=1}^j (R_{nk} \times V_{km}) \quad (3)$$

$$FDM_{mn} = \begin{bmatrix} \sum_{k=1}^j (r_{1k} \times v_{k1}) & \cdots & \sum_{k=1}^j (r_{1k} \times v_{km}) \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^j (r_{nk} \times v_{k1}) & \cdots & \sum_{k=1}^j (r_{nk} \times v_{km}) \end{bmatrix} \quad (4)$$

인력과 개발인력 FDM 행렬을 이용하여 작업에 필요한 개발 인력의 능력을 순위로 정한다. 순위를 정하는 방법은

작업에 대해 FDM 계산의 결과가 높은 것을 1위로 정하고 Equation(5), 같은 경우는 순위를 같게 한다. Equation(6)은 모든 작업에 대해 계산된 Best FDM 행렬이다.

$$Best\ FDM = Rank\ of\ FDM_{mn}\ by\ row\ m \quad (5)$$

$$Best\ FDM_{mn} = \begin{bmatrix} Rank\ of\ FDM_{11} & \dots & Rank\ of\ FDM_{1n} \\ \vdots & \ddots & \vdots \\ Rank\ of\ FDM_{m1} & \dots & Rank\ of\ FDM_{mn} \end{bmatrix} \quad (6)$$

계산된 Best FDM 행렬을 이용하여 각 인력의 평균 FDM(Average FDM)를 Equation(7)과 같이 계산한다. 개발 인력이 가지고 있는 종합능력은 평균 FDM을 가지고 평가한다.

$$Average\ FDM_{mn} = \frac{\sum_{n=1}^m Rank\ of\ FDM_{mn}}{m} \quad (7)$$

개발 인력이 가지고 있는 능력을 우선순위를 아래 Equation(8)와 같이 한다. 개발 능력의 우선순위가 높은 것은 다른 개발 인력에 비해 높음을 나타낸다.

$$Rank\ FDM = [Rank\ of\ Average\ FDM_1 \dots Rank\ of\ Average\ FDM_n] \quad (8)$$

3.3 작업 난이도 순위 계산하기

개발하고자 하는 S/W를 분해한 작업의 난이도를 계산할 필요가 있다. 작업 난이도는 작업이 요구하는 능력 점수의 합을 이용한다. 작업에서 요구하는 능력 요구 점수의 합이 높으면 작업이 어렵다고 볼 수 있다. 이 작업 난이도 계산 방법은 Equation(9)와 같다.

$$Difficulty = \left[\sum_{k=1}^n r_{1k} \dots \sum_{k=1}^n r_{nk} \right] \quad (9)$$

3.4 개발 인력과 작업간의 거리 계산

작업이 요구하는 능력과 개발 인력이 가지고 있는 능력에 대한 거리 계산을 할 필요가 있다. 거리계산은 개발인력 능력과 작업간의 수행 능력을 계산하는 것으로 작업에 최적인 인력을 배치하는데 이용된다. 거리 계산식은 Equation(10)과 같다. 거리계산 행렬은 Equation(11)과 같다.

$$Distance = \sum_{k=1}^m (v_{km} - r_{nk}) \quad (10)$$

$$Distance_{mn} = \begin{bmatrix} \sum_{k=1}^m (v_{k1} - r_{1k}) & \dots & \sum_{k=1}^m (v_{k1} - r_{nk}) \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^m (v_{km} - r_{1k}) & \dots & \sum_{k=1}^m (v_{km} - r_{nk}) \end{bmatrix} \quad (11)$$

3.5 작업에 개발인력 배치하기

협력적인 팀워크를 구성하기 위해 작업에 필요한 개발 인력을 배치한다. 작업에 개발 인력을 배치하는 방법은 Equation(8)의 인력 종합능력 순위와 Equation(9)의 작업 난이도 순위를 이용한다.

우선 계산된 작업 난이도 순위를 이용하여 가장 쉬운 작업을 식별하고, 이 작업에 인력 종합능력 순위 중 가장 약한 개발 인력을 배치한다. 능력이 가장 좋은 개발 인력은 마지막 가장 어려운 작업을 배치한다. 나머지 작업도 작업 난이도와 인력 종합순위를 이용하여 차례대로 배치한다. 이렇게 함으로써 가장 어려운 작업은 능력이 제일 우수한 인력을 배치할 수 있다.

작업에 개발 인력을 배치의 타당성을 검증해야한다. 검증 방법은 3.4절의 거리계산 결과를 이용한다. 거리 계산결과가 0이상이면 작업에 할당된 개발인력이 적절함을 나타낸다. 만약 0미만이면 작업에 할당된 개발 인력이 그 작업에 합당하지 않고, 다른 개발 인력의 도움이 필요함을 나타낸다. 즉 능력이 있는 인력과의 유기적인 협력관계를 통하여 작업을 할 필요가 있음을 나타낸다. 다른 인력과 협력관계를 통해 자신의 능력을 증진하면서 작업을 완성할 수 있다. 만약에 인력 능력과 작업 필요한 능력 차이 많이 나면 인력을 교체해야 되거나 다른 인력이 대신 해주어야 된다.

3.6 개발인력 관계모델 작성하기

에자일 소프트웨어 개발 방법은 유기적인 협력관계를 이용하여 S/W를 개발함으로써, 개발 기간과 생산성을 개선한다. 프로젝트에서 유기적인 협력관계를 형성하기 위해서는 개발 인력간의 관계망 형성이 필요하다. 관계망에서 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 3.2절에서 계산된 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 간선(Edge)으로 연결된다.

개발 인력 관계모델 생성을 위한 두 번째 단계는, 중심노드를 제외한 개발 인력들 간의 연결 관계를 설정한다. 관계를 설정하기 위해 작업-인력 거리 테이블을 이용한다. 인력간의 에지를 만들기 위한 기준 값은 Equation(12)와 같이 가장 능력이 있는 인력의 작업별 거리를 모두 합한 값을 작업의 수로 나눈 후 1의 값을 뺀다. 인력들 간에 에지는 작업-인력 테이블을 이용하여 각 요소의 값이 기준 값 이상인 경우에 설정한다.

$$Standard\ value = \left\lfloor \frac{\sum_{n=1}^n Vmax\ Dis}{n} \right\rfloor - 1 \quad (12)$$

3.7 인력 배치 적절성 검증하기

제시한 인력배치 방안이 팀워크의 효율성을 증진시키는 방안이라는 것을 증명하기 위해 Hao[11]가 말하는 갈등 수준을 이용한다. 갈등수준을 계산하기 위하여 개발인력 관계 모델을 이용하여 갈등수준을 계산한 후, 제시된 방법이 갈등수준의 보통임을 판별한다.

1) Hao[11]의 갈등과 팀의 성능 수준 관계

유기적인 협력관계를 평가하기 위해 Hao[11]가 제안한 갈등 수준에 따른 팀 성과 관계를 활용한다(Table 2). 팀에서 인력들이 서로 교류가 적거나 많으면 갈등수준이 낮거나 높게 나타난다. 갈등수준이 낮거나 높으면 팀의 성과도 낮게 나타난다. 팀에서 적당히 커뮤니케이션이 있으면 갈등수준이 보통이 되고 팀의 효율성과 성과도 더 높게 나타난다[11]. 개발 인력간의 유기적인 협력관계를 평가하기 위해 앞 절에서 할당된 인력과 작업의 관계모델을 만들고 갈등수준을 계산한다.

Table 2. Relationship between the team conflicts and productive

Conflict levels	Low	Normal	High
Team performance	Low	High	Low

2) 갈등수준 계산

갈등수준을 계산하기 위해 개발 인력 관계 모델의 간선(Edge)의 수를 이용한다. 개발 인력관계 모델이 가질 수 있는 최대 간선(Edge)수는 $\frac{n \times (n-1)}{2}$ 개이다. 간선(Edge)수를 이용하여 갈등수준을 계산하기 위해 Equation(13)식 이용한다.

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

간선(Edge)수가 Equation(13)의 범위에 해당할 경우는 갈등 수준이 보통으로 판단하고, Equation(13)의 범위 보다 크거나 작은 경우에는 갈등 수준이 높다고 판정한다.

제안된 모델의 인력 관계모델을 그려 보면 예제가 다음 Fig. 3과 같다.

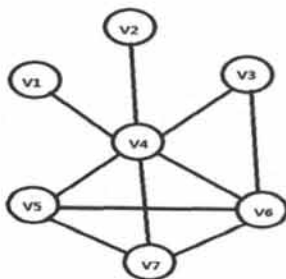


Fig. 3. Example of the relationship model

Fig. 3에서 간선의 수를 계산하면 갈등수준이(7~14범위 내) 보통으로 나타나는 것을 알 수 있다.

그러므로 본 논문에서 제안한 인력 관계 모델을 이용하여 인력을 배치하면 S/W개발 시 유기적인 협력관계가 이루어지게 되어 개발효율과 생산성을 향상시킬 수 있다.

4. 사례 연구

팀워크 관계모델을 적용한 프로젝트 사례를 살펴본다.

4.1 작업 요구 능력 및 팀원 능력 수집

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

- R1 : 리더 결정
- 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

작업을 수행하는데 필요한 능력이 5가지였다. 작업이 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

작업이 필요로 하는 각각의 능력을 5점 척도(1: 나쁨 ~ 5: 아주 좋음)를 이용하여 값을 표시한다. 프로젝트의 7개 작업에 각각의 필요한 능력 값은 아래 Table 3과 같다.

Table 3. Job skills necessary

Ability / Task	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 V1 ~ V7로 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 Table 4와 같다.

Table 4. Human capacity

Human Ability	V1	V2	V3	V4	V5	V6	V7
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

4.2 FDM계산 - 인력 능력 순위

인력의 능력을 순위를 계산하기 위해 FDM계산하는데 이때 Equation(3)을 이용한다. 한 예로 작업 R1에서 인력 V1, V2의 FDM계산은 아래와 같다.

$$FDM_{R1V1} = \sum_{k=1}^5 (R_{1k} \times V_{k1})$$

$$FDM_{R1V1} = (3 \times 5 + 3 \times 4 + 4 \times 3 + 3 \times 4 + 2 \times 4) = 59$$

$$FDM_{R1V2} = (1 \times 5 + 3 \times 4 + 4 \times 3 + 3 \times 4 + 3 \times 4) = 53$$

위의 과정을 이용하여 작업 R1에서 FDM을 계산 하면 RF1=(59 53 59 66 67 83 73) 로 볼 수 있다. 모든 작업의 FDM계산 결과는 아래 Table 5와 같다.

Table 5. FDM calculation results of tasks

	V1	V2	V3	V4	V5	V6	V7
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 Equation(5)을 이용하여 작업에 대한 인력의 순위를 정한다. 모든 작업에 대한 Best FDM (각 작업에서 인력의 우선순위)결과는 Table 6과 같다.

Table 6. Priority of the personnel in the task

	V1	V2	V3	V4	V5	V6	V7
BRF1	5	7	5	4	3	1	2
BRF2	5	7	6	3	3	1	2
BRF3	5	6	6	2	4	1	2
BRF4	5	6	6	3	4	1	2
BRF5	5	7	5	2	4	1	2
BRF6	6	6	5	2	3	1	4
BRF7	5	7	6	3	3	1	2

Table 6의 작업에서 인력의 우선순위를 이용하여 인력 종합능력을 계산한다. 계산하는 방법은 Equation(7)을 이용한다. 각 인력의 Average FDM값은 Table 7과 같다. 이때 Average FDM값이 1에 가까울수록 능력 좋음을 나타낸다. 인력 중 능력이 가장 출중한 인력은 V6임을 알 수 있다.

$$AFDM_{V1} = \frac{5 + 5 + 5 + 5 + 5 + 6 + 5}{7} = 5.14286$$

Table 7. Mean value of the FDM priority

	V1	V2	V3	V4	V5	V6	V7
AFDM	5.142	6.571	5.571	2.714	3.428	1	2.285
DM	86	43	43	29	57	1	71

그래서 인력 능력 순위는: V2<V3<V1<V5<V4<V7<V6

4.3 Difficulty계산 - 작업 난이도 순위

Equation(9)을 이용하여 각 작업의 난이도를 계산 한다. 계산된 작업난이도의 결과는 Table 8과 같다. 작업 난이도의 순위를 표시하면 R6 < R4 = R7 < R3 < R5 < R2 < R1과 같다.

$$Dif1=(5+4+3+4+4)=20$$

Table 8. Difficulty of the each task

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

4.4 Distance 거리 계산

Equation(10)을 이용하여 인력(V)과 작업(R)간의 거리를 계산한다.

$$Distance_{V1R1} = \sum_{k=1}^5 (v_{k1} - r_{1k})$$

$$Dis_{V1R1} = (3-5) + (3-4) + (4-3) + (3-4) + (2-4) = -5$$

작업 R1과 모든 인력의 거리 RD1= -5, -6, -5, -3, -3, 1, -2 이다. 모든 작업과 모든 인력의 거리는 Table 9와 같다.

Table 9. Job - Manpower distance results

	V1	V2	V3	V4	V5	V6	V7
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

4.5 작업에 개발인력 배치하기

작업에 인력을 배치하기 위해서 먼저 쉬운 작업에 낮은 능력을 가진 인력을 배치한다. Table 7과 Table 8을 이용하여 배치를 하면 작업 R6에 인력 V2가 배치된다. 마찬가지로 작업 순위가 그 다음 순위인 R4과 R7과 같이 공동일 경우 능력이 낮은 V3과 V1을 우선 선택한 후, 거리 계산을 한다. 계산된 결과를 이용하여 작업에 배치를 하면 작업 R4에 인력 V3을 작업 R7에 V1을 배치한다. 이런 과정을 거쳐 만들어진 인력-작업 배치결과는 Table 10과 같다.

Table 10. Personnel - work placement results

Workforce skills ranking	V2	V3	V1	V5	V4	V7	V6
Task difficulty ranking	R6	R4	R7	R3	R5	R2	R1

- V1--R7:회사 오피스 작업
- V2--R6:인적 자원 관리 시스템 작업
- V3--R4:재무 관리 시스템 작업
- V4--R5:보관 및 운송 관리 시스템 작업
- V5--R3:해의 무역 업무 관리 시스템 작업
- V6--R1:리더 결정 작업
- V7--R2:프로젝트 조직 및 관리 작업

4.6 팀 조직 협력 관계 모델 만들기

4.2절에서 계산된 개발인력의 능력치 중 가장 높은 능력 값을 가진 개발 인력은 V6이다. 선정된 프로젝트 리더는 중심노드로 다른 개발 인력과 모두 간선(Edge)로 연결된다.

관계를 설정하기 위해 작업-인력 거리 테이블 Table 11을 이용한다. 인력간의 에지를 만들기 위한 기준 값은 $\left\lceil \frac{1+3+7+8+6+9+8}{7} \right\rceil - 1 = 5$ 계산한다. 인력들 간에 에지는 작업-인력 테이블을 이용하여 각 요소의 값이 기준 값 이상인 경우는 인력과 작업 할당인력은 연결한다. 해당되는 인력은 연결 모델은 아래 Fig. 4과 같다.

본 논문에서는 갈등수준 계산하기 위해 개발 인력 관계 모델의 간선(Edge)수를 이용한다. 개발 인력관계 모델이 가질 수 있는 최대 간선(Edge)수는 $\frac{7 \times (7-1)}{2} = 21$ 개가 된

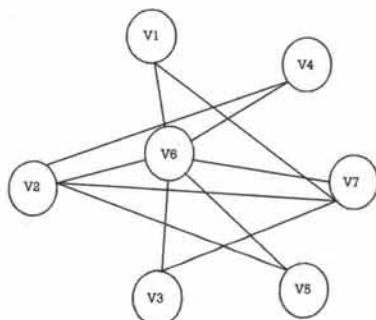


Fig. 4. Human relationship model

Table 11. Workforce-task distance table

	V1	V2	V3	V4	V5	V6	V7
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

다. 간선(Edge)수를 이용하여 갈등수준을 계산하는 식은 다음과 같다.

$$\frac{7 \times (7-1)}{2} \pm \left[\frac{7}{2} \right]$$

그렇다면 간선(Edge)수가 7~15의 범위에 해당할 경우는 갈등 수준이 보통으로 판단한다. 위 Fig. 4에서 간선은 11 때문에 갈등수준이 보통이고 효율성 높고 팀 성과가 일어난다.

5. 결 론

소프트웨어 규모가 점점 커짐으로 인하여 S/W 프로젝트 개발 방식에 팀워크 방식이 적용되고 있다. 좋은 팀 관리, 효과적인 팀워크를 통한 소프트웨어를 개발은 소프트웨어 회사의 피할 수 없는 길이다. 본 논문은 프로젝트 개발에 투입되는 인력과 프로젝트 작업간의 관계모델 형성을 통해, 개발 인력간의 유기적인 협력관계를 형성할 수 있는 기초를 제공하였다. 개발 인력간의 갈등 수준을 낮춤으로써, 대화와 협력을 통해 프로젝트를 진행하게 함으로써 S/W개발의 생산성과 품질을 개선하는데 일조를 하였으며, 개발의 효율을 향상시켰다.

향후 본 연구를 확대하여 프로젝트 관리 도구에 결합시킴으로써, 프로젝트를 담당하고 있는 실제 업무에 프로젝트 관리 효율성을 증진시키고자 한다.

참 고 문 헌

- [1] Seiyong Lee, Hwan-Seung Yong. Design and Evaluation of Agile Framework for Small Projects.
- [2] A Guide to the Project Management Body of Knowledge 4th, 2008.
- [3] Hanseok Ryu, Graduate of Computer Science and Technology, Korea University, A Priority Decision Technique of Problem for Project Management.

- [4] Martin Fowler, Beck, Kent; et al. (2001). "Manifesto for Agile Software Development"
- [5] "Extreme Programming", Computerworld (online), December 2001, webpage: Computerworld-appdev-92
- [6] Coad, P., Lefebvre, E. & De Luca, J. (1999). Java Modeling In Color With UML: Enterprise Components and Process. Prentice Hall International.
- [7] Li,J, Zhao,Q, Chen.S. Simulation robot soccer team collaborative approach. Computer Science, 2002, 29(10): pp.46-48.
- [8] Hu.X, Wang.X. The six elements of the software project management. Computer and Digital Engineering, 2006, 34(5):pp.140-143.
- [9] Ren,J, Wang.E. Interdependence and teamwork. Advances in Psychological Science. 2007, 15(1) pp.146-153.
- [10] Wang.Y, Zhao.X. Teamwork optimal incentive model. Northeastern University (Natural Science), 2007, 28 (9):pp.1346-1349.
- [11] Hao,D, Liu.M. Technology Management Research based on the structure of the research team management of team cohesion. 2005:pp.87-89.



Chen Xiang

e-mail : cxbsr2006@gmail.com

2011년 우송대학교 게임멀티미디어전공
(학사)

2011년~현 재 전남대학교 전자컴퓨터
공학과 대학원생

관심분야: 멀티미디어, 정보처리응용



김 병 기

e-mail : bgkim@jnu.ac.kr

2000년 전북대학교(이학박사)

1996년~1999년 한국정보처리학회 부회장

2007년 한국정보처리학회 제12대 회장

1981년~현 재 전남대학교 전자공학부
교수

관심분야: 웹 사이언스