

제한 시간을 고려한 품질 기반 소프트웨어 프로젝트 계획 생성 기법 * (Quality Based Software Project Staffing and Scheduling with Time Bound)

서동원, 신동환, 배두환

한국과학기술원 전산학과
대전광역시 유성구 대학로 291
{dwseo, donghwan, bae}@se.kaist.ac.kr

요약: 본 연구에서는 소프트웨어 프로젝트 계획에 따른 소프트웨어 품질 점수를 정의하고, 소프트웨어 품질과 동시에 전체 일정 목표를 고려 할 수 있는 프로젝트 계획 생성 기법을 제안한다.

을 수행한 뒤, 주어진 인력할당 결과를 바탕으로 일정을 산출하고, 전체 일정 및 인력 할당 결과에 따른 소프트웨어 품질을 평가한다. 평가는 품질 점수와 시간을 적합성 함수로 사용한다.

핵심어: 소프트웨어 프로젝트 계획, 소프트웨어 프로젝트 관리, 소프트웨어 품질, 인력 할당

1. 서론

비합리적인 프로젝트 계획은 소프트웨어 프로젝트의 실패로 이어질 수 있기 때문에[1], 프로젝트 계획 초기 단계에서 자동화된 프로젝트 계획 생성 기법을 적용해 관리자가 합리적인 프로젝트 계획을 생성할 수 있도록 도와주는 것이 필요하다. 본 연구에서는 프로젝트 계획 생성 요소에 대한 모델을 정의하고(2 장), 소프트웨어의 품질을 정량적으로 정의하여 제한 시간을 고려한 탐색 알고리즘에 적용하는 기법을 설명하고(3 장), 제안된 기법에 대한 평가를 수행한다(4 장).

2. 모델 정의

프로젝트 계획 생성을 위한 개발자, 작업, 생산성 모델을 각각 정의한다. 각 모델은 알고리즘화 가능한 수준으로 자세하게 작성되었으나, 본 논문에서는 간략하게 의미를 서술한다.

개발자 모델은 각 개발자가 갖고 있는 기술에 대한 정보를 포함한다. 한 개발자는 여러 기술을 가질 수 있다[2]. 작업 모델은 프로젝트를 구성하는 작업의 선-후행 관계, 공수, 심각도, 요구되는 기술 집합, 최대 할당 가능한 개발자 수를 포함한다. 생산성 모델은 개발자가 단위 시간 동안 수행할 수 있는 작업의 공수를 계산하는 모델이다. 개발자의 능력과 작업에 투입된 개발자의 인원수에 따라서 생산성이 달라지게 된다.

3. 소프트웨어 품질 기반 프로젝트 계획 생성 기법

본 연구에서는 이전 연구[3]에서 사용한 유전자 알고리즘 기반의 휴리스틱 탐색을 이용하여 인력 할당

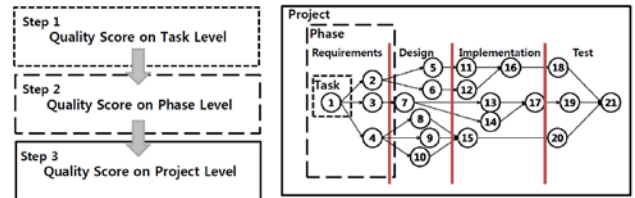


그림 1 소프트웨어 품질 점수 산출 단계

3.1 소프트웨어 품질 점수 산출

소프트웨어 품질의 정량화를 위하여 [4]에서 도출된 산업체 요구사항을 기반으로 집중도, 이해도, 전문성을 정의한다. 집중도 점수는 작업에 할당된 개발자들의 다중 업무 할당 정도를 나타낸다. 이해도 점수는 작업에 할당된 개발자들 중에 해당 작업의 선행 작업에 할당되었던 개발자들의 비중을 나타낸다. [그림 1]와 같이 품질 점수에 각 작업 별, 각 개발 단계 별 특성(작업의 심각도, 단계별 가중치)을 반영하기 위해 단계별로 소프트웨어 품질 점수를 산출한다.

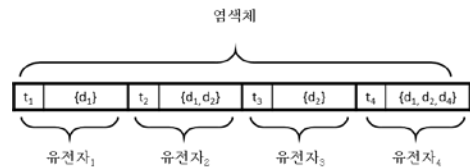


그림 2 유전자 알고리즘의 계획 생성 솔루션 형태

3.2 유전자 알고리즘 기반 계획 생성

유전자 알고리즘은 탐색 알고리즘으로서, 무작위로 솔루션을 생성한 뒤 적합성 함수의 평가에 따라 평가하여 다음 세대에 전이할지를 결정하는 과정으로 수행된다. [그림 2]와 같이 각 유전자는 해당 작업 (t_{id})에 할당된 개발자(d_{id}) 집합을 갖고 있다.

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업[10044457, 자율지능형 지식/기기 협업 프레임워크 기술 개발]과 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)[10041313, UX 지향 모바일 SW 플랫폼 개발]의 일환으로 수행하였음.

3.3 적합성 함수

평가를 위하여 제한시간과 산출된 기간의 관계에 따라서 서로 다른 적합성 값을 사용한다. 제한 시간을 만족한 경우 시간 점수를 1로 유지하되, 제한 시간을 만족하지 못한 경우 시간 점수와 품질 점수를 모두 고려한다. 본 연구에서 제한 시간은 적합성 함수 값을 구하기 위하여 관리자로부터 입력되어야 한다. 만약 입력된 제한 시간이 과도하게 짧아서 해당 제약 조건을 만족할 수 없는 경우 의미 있는 프로젝트 계획 수립이 불가능하다.

4. 사례 연구

총 80개의 실험 데이터 셋에 대하여, 본 연구에서 제안한 기법과 시간만을 고려하는 기법의 프로젝트 계획 결과를 비교하였다.

표 1 시간 기반 기법과의 비교 실험 결과

실험 데이터 셋 (#작업, #개발자)	시간 기반 기법		품질 기반 기법	
	Duration (days)	Quality Score	Success rate	Quality Score
Set1-10(18, 10)	2069	0.59(1)	43%	0.73(1.23)
Set11-20(32, 10)	2785	0.53(1)	77%	0.64(1.22)
Set21-30(32, 15)	2526	0.49(1)	68%	0.64(1.31)
Set31-40(32, 20)	2939	0.45(1)	58%	0.64(1.42)
Set41-50(62, 10)	4446	0.46(1)	99%	0.58(1.25)
Set51-60(62, 15)	3439	0.41(1)	92%	0.51(1.25)
Set61-70(62, 20)	4104	0.37(1)	96%	0.49(1.34)
Set71-80(92, 15)	4743	0.37(1)	99%	0.48(1.30)

[표 1]에서 Success rate은 제안한 기법을 사용한 경우에 대해서, 시간 기반 기법의 최소 시간을 만족하는 솔루션이 나올 확률을 의미한다. 예를 들어, 실험 데이터 셋 1-10에 대해서 Success rate = 43%는 100번의 실험 중 43번의 경우에서 시간 기반 기법의 최소 시간을 만족하는 솔루션이 나왔음을 의미한다.

실험 결과 상으로 제안한 기법은 43% ~ 99% 사이의 Success rate을 보였으며, 품질 점수는 시간 기반 기법보다 23% ~ 40% 더 높은 솔루션을 찾아냈다. 이것은 매우 중요한 발견인데, 기존에 프로젝트 일정만을 고려한 기법과 같은 수준의 최소 시간을 제한 시간으로 만족하면서 동시에 품질을 향상시킬 수 있다는 것을 확인하였다. 특히 실험 데이터 셋 41 이후의 실험 데이터 셋에 대해서는 99%의 높은 Success rate을 보였는데 이는 입력 데이터 셋의 크기가 커질수록, 솔루션의 가능한 탐색 범위가 더 넓어져, 동일한 기간을 가지면서 품질 점수가 더 높은 솔루션을 찾아 낼 수 있었기 때문이다.

5. 관련연구

기존의 소프트웨어 프로젝트 계획 생성 문제는 많은 경우 시간 또는 비용의 최소화만을 다루었다[5]. 두 가지 이상의 목표를 동시에 달성하고자 하는 기법도 제안되었으나[7], 목표에 부합하는 많은 대안 중에서 관리자가 선택해야 하는 어려움이 있다.

6. 결론

본 연구는 프로젝트 계획 생성 기법에서 기존에 프로젝트 일정만을 고려하던 것과 동시에 추가적으로 소프트웨어 품질을 고려할 수 있는 자동화된 프로젝트 계획 생성 기법을 제안하였다. 이를 위하여 프로젝트 계획으로부터 소프트웨어 품질 점수를 도출하는 방법을 정의하였으며, 프로젝트에서 요구하는 시간 조건을 만족하기 위해서 제한 시간이라는 제약 조건을 만족하는 기법을 제안하였다. 사례 연구를 통하여 본 연구에서 제안한 기법이 기존 시간 기반 기법의 최소 시간을 만족함과 동시에 더 높은 품질 점수를 갖는 계획을 생성할 수 있음을 확인하였다. 제한 시간을 활용하여 적합성 함수를 구현함으로써 시간과 품질을 모두 만족하는 프로젝트 계획을 효과적으로 탐색할 수 있었기 때문이다. 뿐만 아니라 품질 점수를 작업, 개발 단계, 프로젝트 수준으로 구성함으로써 작업의 심각성과 DAM(Defect Amplification Model) 모델에 따른 각 개발 단계의 가중치를 부여할 수 있었다.

참고문헌

- [1] R.-G. Ding and X.-H. Jing, Five Principles of Project Management in Software Companies, Project Management Technology, vol. 1, 2003.
- [2] Bellenguez-Morineau, Odile, and Emmanuel N_eron. A branch-and-bound method for solving multi-skill project scheduling problem., RAIRO-Operations Research, vol.41, no.2, pp. 155-170, 2007
- [3] Jihun Park, Dongwon Seo, Gwangui Hong, Donghwan Shin, Jimin Hwa, Doo-Hwan Bae, Practical Human Resource Allocation in Software Projects Using Genetic Algorithm, 26th International Conference on Software Engineering and Knowledge Engineering, pp. 688-694, 2014.
- [4] 화지민, 박지훈, 신동환, 홍광의, 배기곤, 서영석, 배두환, “소프트웨어 산업체 피드백을 반영한 자동화된 인력 할당”, 정보과학회논문지:소프트웨어 및 응용, 제 40 권, 제 7 호, pp.369-380, 2013년 7월.
- [5] Peixoto, Daniela CC, Geraldo R. Mateus, and Rodolfo F. Resende, The Issues of Solving Staffing and Scheduling Problems in Software Development Projects, Computer Software and Applications Conference, 2014.
- [6] Constantinos Stylianou, Andreas S. Andreou, A Multi objective Genetic Algorithm for Software Development Team Staffing Based on Personality Types, Artificial Intelligence Applications and Innovations, Springer Berlin Heidelberg, 2012.