

A Measuring Model of Risk Impact on The App Development Project in The Social App Manufacturing Environment

Baek Jung Hee[†] · Lim Young Hwan^{**}

ABSTRACT

Crowd Sourcing-based Social App Manufacturing environment, a small app development project by a team of anonymous virtual performed without the constraints of time and space, and manage it for the app development process need to be automated method. Virtual teams with anonymity is a feature of the Social App Manufacturing, is an important factor that increases the uncertainty of whether the completion of the project or reduction in visibility of the progress of the project. In this study, as one of how to manage the project of Social App Manufacturing environment, the impact of risk that can be used to quantitatively measure the impact of the risk of delay in development has on the project also proposes a measurement model. Effects of risk and type of the impact of risks associated with delays in the work schedule also define the characteristic function, measurement model that has been proposed, suggest the degree of influence measurement equation of risk of the project in accordance with the progressive. The advantage of this model, the project manager is able to ensure the visibility of the progress of the project. In addition, identify the project risk of work delays, and to take precautions.

Keywords : Social App Manufacturing, App Development Project Management, Project Risk Measurement

Social App Manufacturing 환경의 앱 개발 프로젝트에서 위험영향도 측정 모델

백 정 희[†] · 임 영 환^{**}

요 약

Crowd Sourcing 기반의 Social App Manufacturing 환경에서는 소규모 앱 개발 프로젝트를 익명의 가상팀에 의해 시공간의 제약이 없이 수행하며 이를 위해 앱 개발 관리를 위한 자동화된 방법을 필요로 한다. Social App Manufacturing의 특징인 익명성을 갖는 가상팀은 프로젝트 진행 상황에 대한 가시성의 저하와 프로젝트 완료 여부의 불확실성을 증가시키는 핵심 요인이다. 본 연구에서는 Social App Manufacturing 환경하의 프로젝트 관리 방법의 하나로 개발 지연이 프로젝트에 주는 위험 영향을 정량적으로 측정할 수 있는 위험영향도 측정 모델을 제안한다. 제시한 측정 모델은 작업일정 지연에 따른 위험 영향 유형과 위험영향도 특성함수를 정의하고 진행 성과에 따른 프로젝트 위험영향도 측정식을 제안한다. Social App Manufacturing 환경에서 수행하는 프로젝트 관리에 이 모델을 활용하여 진행 상황에 대한 가시성을 확보하고 일정 지연으로 실패 위험이 발생하는 프로젝트를 식별하고 미리 대처할 수 있도록 한다.

키워드 : 소셜 앱 매뉴팩처링, 소셜 앱 개발 프로젝트 관리, 프로젝트 위험 측정

1. 서 론

인터넷과 모바일 등의 정보통신기술 발달로 외부자원을 내부자원과 결합하여 새로운 서비스 개발 아이디어를 창출

하고 신 서비스 기반의 비즈니스 모델을 개발하는 개방형 협업 모델[1]이 가능하다. 제조분야에서는 제조사는 상품에 대한 아이디어나 디자인만을 개발하며 상품은 전 세계에 있는 공장 중에서 선택하여 생산하는 소셜매뉴팩처링(Social Manufacturing) 기법이 활용되고 있다[2]. 불특정 다수가 참여하는 소셜 네트워크 기반서비스로 소셜 네트워크상에 아이디어를 제시하고 아이디어의 비즈니스 가능성에 따라 제품 개발/제조를 할 수 있도록 자금을 투자하는 크라우드펀딩(Crowd Funding)[3]이나 인터넷상으로 작업에 참여할 사

[†] 준 회원 : 송실대학교 미디어학과 박사과정 수료
^{**} 종신회원 : 송실대학교 미디어학과 교수
Manuscript Received : March 19, 2014
First Revision : May 26, 2014; Second Revision : June 30, 2014
Accepted : July 24, 2014
* Corresponding Author : Baek Jung Hee(junghee.baek@gmail.com)

람을 모집하고 그들에게 처리할 업무를 할당하거나 참여 역할을 주고 그 결과를 취합하는 형태로 전 세계의 누구라도 참여가 가능한 크라우드소싱(Crowd Sourcing)[4] 등이 있다. 크라우드소싱에 참여하는 자는 자신의 여유시간을 활용하여 더욱 많은 수익을 창출하고자 한다[5].

최근의 앱 개발 플랫폼은 개발자가 자신이 보유한 아이디어나 기술을 최대한 발휘하여 수익을 창출할 수 있는 새로운 길을 제공해 준다. 앱 스토어(마켓플레이스)의 성장은 앱 개발에 대한 수요를 증대시킨다. 이러한 앱 개발 수요는 자신의 아이디어와 필요에 의해 앱을 만들고자 하는 고객과 자신의 기술을 발휘하여 수익을 창출하고자 하는 개발자 간의 중개자로서 Social App Manufacturing[6]이 가능하도록 한다. Social App Manufacturing은 소셜네트워크 상에서 앱 개발 주문을 받고 크라우드소싱으로 개발자를 모집하여 프로젝트를 수행한다. Social App Manufacturing에서 수행하는 앱 개발 프로젝트팀은 근무환경에 대한 제약이 없으며 온라인으로만 연결 가능한 익명성을 갖는 개발자로 구성된다. 이런 특징은 개발팀이 동일 장소에서 매일 면대면(face-to-face)으로 작업을 수행하는 기존의 일반적인 프로젝트보다 프로젝트의 진행 상태를 파악하기 어렵게 하며 프로젝트 완료여부에 대한 불확실성을 높인다. 따라서 Social App Manufacturing의 프로젝트의 수행상태에 대한 가시성과 성공 가능성을 높일 수 있는 효율적인 프로젝트 관리 방법이 필요하다.

이를 위해 본 연구에서는 앱 개발에서 업무 분석과 설계 산출물인 스토리보드를 작업 성과 관리에 활용하여 수행과정에서 발생하는 개발 일정 지연이 프로젝트에 주는 위험영향을 정량적으로 측정하는 모델을 제시하고자 한다. 이 모델을 일정관리에 활용하여 자동으로 위험영향도를 측정하여 개별 프로젝트의 상태에 대한 가시성을 확보하고 수행 중인 다수의 프로젝트 중에서 위험한 프로젝트를 식별하여 프로젝트 관리자에게 제시함으로써 개별 프로젝트 관리에 투입되는 오버헤드를 줄일 수 있도록 한다.

2. Social App Manufacturing에서 앱 개발 프로젝트 관리의 문제점

2.1 Social App Manufacturing 모델

다양한 시장의 요구에 대응하고 빠른 시간 내에 제품 출시를 목적으로 하는 Social App Manufacturing은 아웃소싱[7-9]과 제품 생산(개발) 관리 기법을 온라인 상으로 옮겨 놓은 앱 개발 관리 플랫폼이다. 앱 제작 주문에서 마켓 배포까지 발생하는 일련의 앱 제작 과정을 지원하는 자동화된 체계인 앱 개발 벨트(Fig. 1)를 제공한다[10]. 앱 개발은 설계, UI디자인, 코딩 단계로 구성되며 각 단계별 전문 개발자에 의해 수행된다. 개발팀은 지원자 경쟁 공모 방식으로 시스템의 개발자모집벨트에서 자동으로 구성한다. 이들 개발자는 각자 주어진 작업을 수행하고 약속된 산출물을 Social App Manufacturing에서 제공하는 벨트에 탑재한다. 앞 단

계의 산출물은 자동으로 다음 단계로 참여 개발자에게 전달된다. 이 과정이 앱 개발이 완료될 때까지 반복된다.

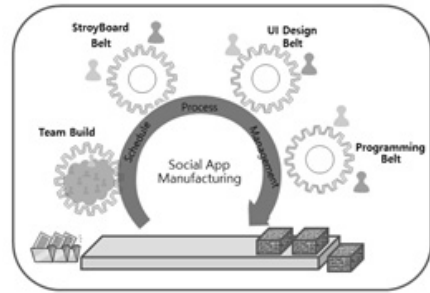


Fig. 1. Social App Manufacturing Model

2.2 관련 연구 및 문제점

Social App Manufacturing 모델이 갖는 특징은 첫 번째, 온라인으로만 운영하며 앱 개발 프로젝트 관리를 자동화하여 오버헤드를 최소화하고자 한다. 두 번째, 시공간의 제약을 받지 않으며 공유된 작업 장소가 없으며 작업 시간의 동시성이 없다. 세 번째, 프로젝트 팀원 모두 익명성을 갖는다. 프로젝트 참여자는 서로를 알지 못하고 만나지도 않으며 개발 결과물만을 제공한다. 네 번째로, 프로젝트는 짧은 기간 동안 수행하며 소수의 개발자가 참여하는 소규모 프로젝트이다.

일반적으로 소프트웨어개발 프로젝트의 관리 방법으로는 프로젝트의 성격에 맞는 소프트웨어개발 방법론[11]과 프로젝트관리 방법을 선택하여 적용한다. PMBOK[12]에서는 프로젝트 3대 제약조건으로 범위, 일정, 비용을 정의하고 이들 요소가 품질에 영향을 준다고 보고 관리하도록 한다. 소프트웨어 개발에서의 위험을 불만족스런 결과물이 나올 가능성 P(UO)와 그로 인한 손실 부분 L(UO)의 관계로 보고 위험노출(Risk Exposure)을 $RE = P(UO) \times L(UO)$ 로 정의하기도 한다[13]. 최근에 대두되고 있는 Agile 기법[14, 15]인 XP, Scrum[16, 17], Lean 소프트웨어개발방법[18] 등은 전통적인 소프트웨어 개발 방법에 비해 협업과 신속한 개발을 중시한다. 이들에서 제시하는 다양한 기법과 비교(Table 1) 하면 코드중심 개발, 잦은 릴리즈 등은 Social App Manufacturing에서 적용 가능하다. 익명성을 갖는 Social App Manufacturing의 가상팀에 대하여 동일 작업 공간이나 Pair Programming, Daily Scrum Meeting과 같은 오프라인 중심의 기법을 적용하기는 매우 어렵다. 또한 프로젝트 기간 중에 익명의 개발자가 개발을 중단할 가능성이 존재함에도 이를 신속히 인지하기 어렵다. Social App Manufacturing 프로젝트 수행 과정에 발생하는 이러한 어려움은 프로젝트 수행 상태에 대하여 파악할 수 없으며 완료에 대한 불확실성을 증대시킨다. 이는 일반적인 프로젝트 관리 방법을 적용하기 어렵게 한다. 따라서 Social App Manufacturing 프로젝트 환경을 고려한 위험 관리 방법을 필요로 한다.

Table 1. Comparison Offline-based Project And Social App Manufacturing-based Project

| | | Offline Based Project | Social App Manufacturing Based Project |
|------------|-----------------------------|--|--|
| Work Style | Work Type | Off-line | Social Based On-line |
| | Team Type | Real project team | Virtual project team |
| | Workplace | Real project office (in the same place/office) | Don't know their workplace (Available anywhere) |
| | Activity Visibility | Visible | Invisible |
| | On-duty time | Regular and the same time | Anytime |
| Team | Acquainted with team member | Yes | No (Anonymous) |
| | Membership | Will be increased during project execution | No membership and No chance to make their membership |
| | Cooperation | Available discussion, meeting and sharing issues daily(frequently) | Difficult |

3. Social App Manufacturing에서 앱 개발 프로젝트 위험영향도 측정 모델

3.1 Social App Manufacturing에서 발생하는 프로젝트 위험 관리

Social App Manufacturing에서 작업 일정 관리에 스토리 보드를 활용한다. 스토리보드는 개발할 내용을 제시하며 그에 따라 화면UI 및 코드 개발이 진행된다. 스토리보드에서 제시된 기능이 구현되었는지가 프로젝트 완료 여부 판단의 근거가 될 수 있다. 개별 화면단위를 업무처리 기준으로 그룹화한 스토리를 구성하고 중요도나 작업량 등을 고려하여 우선순위전략으로 프로젝트 일정계획을 수립하고 짧은 주기로 실행상태를 점검한다. 작업이 계획대로 진행되지 않는 경우 프로젝트가 실패가능성이 높아진다고 볼 수 있다. 프로젝트 관리자에게 프로젝트가 얼마만큼 위험한 상태에 있는지를 측정하여 프로젝트 진행 상태를 정량적으로 판단할 수 있도록 하고, 프로젝트가 더 이상의 위험한 상태가 되지 않도록 하고자 한다. 이를 위해 스토리보드 기반의 앱 개발 프로젝트 위험영향도 측정 모델을 제시한다.

본 연구에서 제시하는 위험영향도 측정 모델은 Social App Manufacturing에서 수행하는 3개월 이하의 2~3명이 참여하는 소규모 앱 개발 프로젝트에 한정하여 적용한다.

3.2 스토리 위험 유형과 위험영향도 특성 함수

프로젝트의 진도 관리 계획을 스토리 단위로 하여 작업 완료 목표를 수립한다. 일반적으로 초기 계획대로 진행되지 않고 지연이 발생하는 경우 실패가능성이 높아진다고 할 수 있으므로 개별 스토리에 대한 지연 정도를 위험 측정을 위한 요소로 사용한다.

스토리별 위험영향도를 측정하기 위해서 스토리 개발이 지연되었을 때 프로젝트에 주는 위험 유형을 찾고 그 위험 유형에 따른 위험영향도 특성함수를 결정한다. 프로젝트에서 구현할 스토리들에 대하여 스토리의 중요도와 개발 일정을 고려하여 위험 유형을 구분한다. 위험유형을 찾기 위해 일반적으로 많이 사용하는 일정 계획을 수립할 때 중요한 부분을 개발 일정 초기에 배치한다는 우선순위 전략을 활용할 수 있다. 초기에 배치된 중요 스토리에 대한 개발이 제대로 진행되지 않는 경우 프로젝트 위험이 초반부터 크게 나타나며 지연이 될수록 그 위험은 더욱 커지게 된다. 프로젝트 후반에 배치된 스토리는 남은 프로젝트 수행 기간이 적으므로 중요도는 낮지만 작은 지연에도 프로젝트 성공 여부에 영향을 준다. 이러한 점을 고려하여 위험유형을 2가지 (Type A, Type B)로 구분한다. 각 유형의 스토리 개발이 지연되었을 때 프로젝트에 주는 위험영향도는 Fig. 2와 같은 형태로 나타나게 된다. 프로젝트 기간을 E, 수행 경과 기간을 d라고 할 때, 스토리별 위험 유형에 따른 위험영향도 특성 함수를 다음과 같이 정의한다.

- 스토리 위험 유형 Type A의 위험영향도 특성함수($f_A(d)$) 프로젝트 내에서 상대적으로 중요도가 낮으며 주로 프로젝트 중반 이후에 개발 일정이 배정된다.

$$f_A(d) = \left(\frac{d}{E}\right)^2 \tag{1}$$

- 스토리 위험유형 Type B의 위험영향도 특성함수($f_B(d)$) 프로젝트의 핵심 아이디어 또는 기술의 구현을 포함하는 중요한 스토리이다. 프로젝트 완료에 영향을 많이 주며, 주로 프로젝트 개발 일정 초기에 배정된다.

$$f_B(d) = 1.35 \times \ln\left(1 + \frac{d}{E}\right) \tag{2}$$

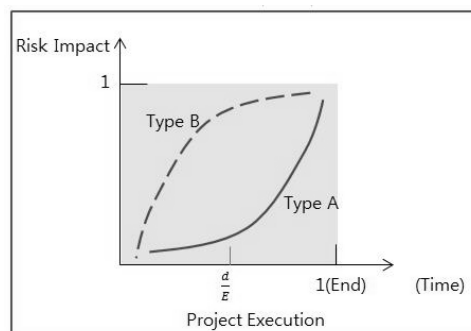


Fig. 2. Type of risk according to the characteristics of the story

3.3 위험영향도 측정 모델

Social App Manufacturing의 프로젝트의 개발 지연에 의한 위험영향도를 측정하기 위해 스토리별 위험영향도를 측정한다.

- 프로젝트 총 수행기간(E)
- (n-1) 번째 중간점검시점과 (n) 번째 중간 점검 시점 사이의 수행경과 기간 : d_n
- (n) 번째 중간 점검 시점까지 계획된 작업량 : $O_e(d_n)$
- (n) 번째 중간 점검 시점까지 달성한 작업량 : $O_a(d_n)$

프로젝트 종료 시점(E)까지 완료할 작업량 : $O_e(E) = 100$ 이라 할 때, 스토리별 위험영향도($r(d)$)는 다음과 같다.

$$r(d) = \begin{cases} 1 & \text{if } d_N = E \text{ and } O_a(d_n) \neq O_e(d_n) \\ U(d) \times f(d) & \text{if } d_n < E \end{cases} \quad (3)$$

$$U(d) = \frac{O_e(d) - O_a(d)}{O_e(d)} \quad (4)$$

- 위험영향도 특성 함수 : $f(d)$
- 중간 점검 시점(d)에서 측정된 미달성율 : $U(d)$

이때, 중간 점검 시점에서의 프로젝트 전체에 대한 위험영향도(또는 프로젝트 위험영향도) R(d)를 정의할 수 있다. 스토리(i)의 위험영향도 $r_i(d)$ 는 위험영향도 특성함수 값과 미달성률에 비례하므로 가장 큰 위험이 프로젝트 실패에 주는 영향은 다른 여러 개의 작은 위험들보다 더 크다고 볼 수 있으므로 중간 점검 시점(d)에서의 개별 스토리에 대한 위험영향도 $m(d)$ 중 최대값을 그 시점에서의 프로젝트 전체에 대한 위험영향도로 정한다.

$$R(d) = \text{Max}(r_1(d), \dots, r_i(d), \dots, r_N(d)) \quad i = 1 \dots N \quad (5)$$

4. Social App Manufacturing에서 앱 개발 프로젝트의 위험영향도(R) 측정 모델 적용

4.1 예제 프로젝트

3절에서 제시한 프로젝트 위험영향도 측정 모델을 App Factory에서 수행한 프로젝트에 적용하여 프로젝트의 위험영향도를 시뮬레이션 하였다.

예제 프로젝트는 App Factory에서 수행한 스마트폰에서 사진을 촬영하고 SNS상으로 공유하는 서비스를 제공하는 스마트폰용 앱 개발 프로젝트이다. 이 앱의 스토리보드는 21개의 화면이 있는 4개의 스토리로 구성된다. 프로젝트 수행기간은 총 40일 동안 수행하며, 10일 경과 시점마다(총 4회의 중간 점검) 개발 달성량을 보고하도록 설정하였다. 스토리당 작업량은 각 구간별로 0과 100 사이에서 할당하며 누적량이 100이면 완료이다. AppFactory의 개발자선정벨트에서 선정된 개발자들은 개발 대상인 스토리와 일정(중간 점검일 및 수행목표) 등을 할당받으며 중간 점검 시점이 되면 자신의 개발 성과를 App Factory의 개발 벨트에 보고한다. 보고된 진척 상태를 기반으로 각 점검 시점의 프로젝트 위험영향도를 측정한다. 측정된 위험영향도는 프로젝트 관리자에게 통보되며, 그에 따라 위험 관리를 수행한다.

4.2 프로젝트 위험영향도 측정

예제 프로젝트의 작업 일정 결과에 따른 위험영향도 측정은 다음과 같은 단계에 따라 진행한다.

1) 스토리별 위험영향도 유형 식별 및 일정 계획 수립

먼저 프로젝트의 구현 대상 스토리의 위험 유형을 프로젝트 내에서의 중요도 등을 고려하여 결정한다. 프로젝트 중간 점검 시점에서의 작업 달성해야 할 목표량을 Table 2와 같이 계획한다.

Table 2. Story Risk Types and Planned Outputs

| Story ID | Story Risk Type | Project Check Point | | | | | Total Output |
|----------|-----------------|---------------------|-----|-----|-----|-----|--------------|
| | | Start | 1st | 2nd | 3rd | End | |
| PC_S1 | A | 0 | 0 | 0 | 75 | 25 | 100 |
| PC_S2 | B | 0 | 50 | 50 | 0 | 0 | 100 |
| PC_S3 | B | 0 | 50 | 50 | 0 | 0 | 100 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 100 | 100 |
| Sum | | 0 | 100 | 100 | 75 | 125 | 400 |

2) 스토리별 작업 진행 결과 점검

- Case 1 : 초기 일정 계획대로 원만히 진행되며 프로젝트 기간 내에 개발이 완료된 성공한 경우로 스토리별 중간 점검 시점에서의 성과는 Table 3과 같다.

Table 3. An Estimated Output of Project(Case 1)

| Story ID | Story Risk Type | Project Check Point | | | | | Total Output |
|----------|-----------------|---------------------|-----|-----|-----|-----|--------------|
| | | Start | 1st | 2nd | 3rd | End | |
| PC_S1 | A | 0 | 0 | 0 | 75 | 25 | 100 |
| PC_S2 | B | 0 | 50 | 50 | 0 | 0 | 100 |
| PC_S3 | B | 0 | 50 | 50 | 0 | 0 | 100 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 100 | 100 |
| Sum | | 0 | 100 | 100 | 75 | 125 | 400 |

3) 프로젝트 위험영향도 산정

작업 계획(Table 1)과 작업 결과(Table 2) 데이터를 식 3과 식 5에 적용하여 스토리별 위험영향도 $r_n(d)$ 와 프로젝트의 위험영향도(R)를 측정한다(Table 4). 그 결과, 모든 스토리가 중간 점검 시점의 계획대로 작업량을 달성하였으므로 스토리별 위험영향도는 $r_n(d) = 0$ 이 되었다.

Table 4. Measured Risk Impact on the Project Check Point(Case 1)

| Story ID | Story Risk Type | Story Risk Impact | | | | |
|------------------------|-----------------|-------------------|-----|-----|-----|-----|
| | | Start | 1st | 2nd | 3rd | End |
| PC_S1 | A | 0 | 0 | 0 | 0 | 0 |
| PC_S2 | B | 0 | 0 | 0 | 0 | 0 |
| PC_S3 | B | 0 | 0 | 0 | 0 | 0 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 0 |
| R(Project Risk Impact) | | 0 | 0 | 0 | 0 | 0 |

4) 프로젝트 진행 상황을 달리한 경우 위험영향도 측정

프로젝트 진행 상태에 따른 위험영향도 비교를 위해 앞의 예제 프로젝트에 대하여 프로젝트 진행 상황을 달리하여 위험영향도를 측정하였다.

- Case 2 : 초기 계획된 일정에 완료된 작업 양이 미달한 적이 있지만 최종적으로는 납기일 내에 개발이 완료된 성공한 경우로 중간점검시점에 완료한 작업량은 Table 5와 같다.

Table 5. An Estimated Output of Project(Case 2)

| Story ID | Story Risk Type | Project Check Point | | | | | Total Output |
|----------|-----------------|---------------------|-----|-----|-----|-----|--------------|
| | | Start | 1st | 2nd | 3rd | End | |
| PC_S1 | A | 0 | 0 | 0 | 55 | 45 | 100 |
| PC_S2 | B | 0 | 40 | 40 | 20 | 0 | 100 |
| PC_S3 | B | 0 | 40 | 20 | 10 | 30 | 100 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 100 | 100 |
| Sum | | 0 | 80 | 60 | 85 | 175 | 400 |

Table 5의 작업 완료 결과를 작업 계획(Table 2)에 대하여 측정된 위험영향도는 Table 6과 같다. 스토리 PC_S1(스토리위험유형 Type A)는 3번째 중간 점검에서 목표대비 작업량 20을 미달성 하였으며 위험영향도 $r_n(d)=0.15$ 가 산출된다. 이 시점에서 프로젝트 전체의 위험영향도(R)는 스토리별 위험영향도 중 최대값인 0.2815이다.

Table 6. Measured Risk Impact on the Project Check Point(Case 2)

| Story ID | Story Risk Type | Story Risk Impact | | | | |
|------------------------|-----------------|-------------------|---------|---------|---------|-----|
| | | Start | 1st | 2nd | 3rd | End |
| PC_S1 | A | 0 | 0 | 0 | 0.15 | 0 |
| PC_S2 | B | 0 | 0.06025 | 0.18246 | 0 | 0 |
| PC_S3 | B | 0 | 0.06025 | 0.36492 | 0.28125 | 0 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 0 |
| R(Project Risk Impact) | | 0 | 0.06025 | 0.36492 | 0.28125 | 0 |

- Case 3 : 초기 계획된 일정에 비하여 빈번하게 작업 지연이 발생하였으며 최종적으로 납기일 내에 개발이 완료되지 않은 경우이다. 중간 점검 시점에서 확인된 작업 완료량은 Table 7과 같다.

Table 7. An Estimated Output of Project(Case 3)

| Story ID | Story Risk Type | Project Check Point | | | | | Total Output |
|----------|-----------------|---------------------|-----|-----|-----|-----|--------------|
| | | Start | 1st | 2nd | 3rd | End | |
| PC_S1 | A | 0 | 0 | 0 | 50 | 40 | 90 |
| PC_S2 | B | 0 | 40 | 40 | 20 | 0 | 100 |
| PC_S3 | B | 0 | 40 | 20 | 10 | 10 | 80 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 100 | 100 |
| Sum | | 0 | 80 | 60 | 80 | 150 | 370 |

Table 7의 완료 작업량으로 측정된 위험영향도는 Table 8과 같다. 스토리 PC_S3은 두 번째 중간 점검 시점에는 완료되어야 하나, 첫 점검 시점에서는 10, 두 번째는 40, 세 번째는 30만큼의 작업을 미달성하고 최종적으로 작업량 20을 미달성하고 종료하였다. 이 스토리의 경우 작업 미달성 상태가 계속 발생하여 미달성 작업량이 누적되고 중간 점검 시점마다 측정된 위험영향도도 증가하고 있다.

Table 8. Measured Risk Impact on the Project Check Point(Case 3)

| Story ID | Story Risk Type | Story Risk Impact | | | | |
|------------------------|-----------------|-------------------|---------|---------|---------|---------|
| | | Start | 1st | 2nd | 3rd | End |
| PC_S1 | A | 0 | 0 | 0 | 0.1875 | 0.2 |
| PC_S2 | B | 0 | 0.06025 | 0.18246 | 0 | 0 |
| PC_S3 | B | 0 | 0.06025 | 0.36492 | 0.42188 | 0.66667 |
| PC_S4 | A | 0 | 0 | 0 | 0 | 0 |
| R(Project Risk Impact) | | 0 | 0.06025 | 0.36492 | 0.42188 | 0.66667 |

5) 프로젝트 진행 상황별 위험영향도 측정 결과 분석

앞의 세 가지 프로젝트 진행상황에 따라 측정된 위험영향도 결과를 살펴보면, Case 1은 모든 계획과 실행 작업량이 일치하여 성공적으로 종료된 프로젝트 유형으로 프로젝트의 수행 과정의 모든 중간 점검 시점에서 위험영향도가 모두 0으로 측정되었으며, 따라서 중간 점검 시점에서 위험은 없는 것으로 나타난다. Case 2에서는 프로젝트 수행 과정에 중에 일부 지연이 발생하였으나 개발을 완료한 프로젝트 유형으로 중간 점검 시점에서 계획 대비 지연이 발생하여 해당 시점에 위험영향도가 증가하는 것으로 나타났다. 다음 점검 시점에서 미진한 부분을 만회하여 위험영향도를 줄이고 최종적으로 모든 개발을 완료하였으며, 최종 시점에서 위험영향도는 0으로 나타난다. 중간 과정의 지연 부분에 대하여 위험 영향도가 측정되어 위험이 존재함을 알 수 있고 위험의 변화부분도 파악할 수 있다. Case 3은 프로젝트 수행과정에 지연이 심각하였으며 최종적으로 고객에게 프로젝트 산출물을 납품하지 못하고 프로젝트가 실패로 종료된 경우이다. 이 프로젝트 유형에서는 중간 시점에서 계획 대비 실적이 부진하였으므로 그 상황이 그대로 위험영향도로 측정되었다. 또한 프로젝트가 종료되어야 하는 시점에서도 개발 대상을 모두 완료하지 못하였으므로 프로젝트 최종 마감 시점에도 위험영향도가 크게 측정되었다.

프로젝트 진행 상황을 달리하여 위험영향도를 측정된 결과, 진행 결과의 지연 정도에 따라 위험영향도를 측정할 수 있었으며 지연이 지속되는 경우 위험영향도가 증가하고, 일정이 만회되면 위험영향도도 감소하는 것을 알 수 있다. 중간 점검 시점에 측정된 진척상황으로부터 자동으로 위험영향도를 산출하여 프로젝트 관리자에게 통보하면 프로젝트 관리자는 별도의 노력을 기울이지 않아도 프로젝트의 상태를 변화를 파악할 수 있고, 어떤 스토리가 해당 프로젝트에 위험영향도를 크게 주는지 알 수 있어 그에 따라 해당 스토리를 개발하는 개발자에 대하여 일정 지연에 따른 조치를 취할 수 있다.

5. 결 론

본 논문에서는 Social App Manufacturing 환경하에서 수행하는 앱 개발 프로젝트 관리에 활용할 수 있도록 작업 일정 지연이 프로젝트에 주는 위험영향도 측정모델을 제시하였다. 이 위험영향도 모델에서는 프로젝트를 구성하는 스토리의 속성(중요도, 개발 일정상의 우선순위 등)을 고려한 위험유형을 정의하고 이 위험유형에 따라 스토리 개발일정 지연이 프로젝트에 주는 위험 영향을 측정하도록 하였다. 또한 예제 프로젝트에서 측정된 수행 상태 외에 프로젝트 수행 상태를 달리하여 측정된 위험영향도를 비교하였다.

Social App Manufacturing 환경은 앱에 대한 수요를 만족하고 기술 인력 부족을 해결할 수 있는 새로운 아웃소싱 비즈니스 형태를 제시한다. 그러나 Social App Manufacturing 환경이 주는 이 장점들은 오프라인 중심의 앱 개발 프로젝트와 비교할 때 수행 상태 파악이나 개발자 관리(익명성, 비상주, 부업 형태)를 어렵게 하며, 이는 납기일 내에 프로젝트를 완료할 수 있는가에 대한 불확실성을 증가시킨다. 따라서 본 논문에서는 일정관리에 중점을 두고 Social App Manufacturing 프로젝트에 중간점검을 적용하여 개발진척을 보고하도록 프로세스를 개선하고 이를 활용하여 프로젝트 성공에 주는 위험영향을 파악할 수 있도록 하였다.

향후 연구에서는 Social App Manufacturing 환경에서 수행하는 다양한 앱 개발 프로젝트를 대상으로 제시한 위험영향도 모델을 적용한 사례를 비교하여 위험영향도 모델의 효용성을 연구하고자 한다. 본 논문에서 고려한 앱 유형 외에 다양한 앱 유형에 대하여 프로젝트 위험영향도 측정 모델을 적용할 수 있도록 확장 연구가 필요하다. 또한, Social App Manufacturing에 참여하는 앱 개발자에게 프로젝트 일정 준수를 촉진할 수 있는 인력 및 일정관리 방안에 대한 연구와 또한 위험이 발생된 프로젝트를 Social App Manufacturing 환경에서 효율적으로 관리할 수 있는 방안에 대한 연구가 필요하다.

References

[1] Z. Lee, M. Lee, and Y. Chu, "Open Collaboration Innovation Methodology (OCIM): A Methodology for New Service Development," Journal of Society for e-Business Studies, Vol.16, No.1, pp.49-70, 2011.

[2] MFG.com, <http://www.mfg.com> [Internet], <http://www.mfg.com>

[3] Mollick, Ethan, "The dynamics of crowdfunding: An exploratory study," Journal of Business Venturing, Vol.29, No.1, pp.1-16, 2014.

[4] Quirky, <http://www.quirky.com> [Internet], <http://www.quirky.com>

[5] J. Howe, The Rise of Crowdsourcing [Internet], <http://www.wired.com/wired/archive/14.06/crowds.html>

[6] 7GNC, <http://www.appbelt.co.kr> [Internet], <http://www.appbelt.co.kr>

[7] D. H. Cho, "Factors Affecting the Project Performance of Software Development Outsourcing," Journal of the Korea Academia-Industrial, Vol.13, No.3, pp.1077-1083, 2012.

[8] J. Kwak and J. An, "A Study of the Core Success Factors for the Development of Application Offshore," Entru Journal of Information Technology, Vol.12, No.2, pp.39-53, 2013.

[9] J. N. Lee, "Business Strategy, Outsourcing and Firm Performance: An Exploratory Study from a configurational Perspective," Korean Management Review, Vol.34, No.5, pp. 1555-1583, 2005.

[10] E. Y. Kim, "A Study on the App Production Belt Model and the Revenue-Sharing Scheme for building Social Network based App Factory," Ph.D. dissertation, Soongsil University, Seoul, Korea, 2012.

[11] R. S. Pressman, Software Engineering : Practitioner's Approach, 7th ed., New York: McGraw-Hill Higher Education, 2009.

[12] Project Management Institute, PMBOK: A Guide to the Project Management Body of Knowledge, 5th Edition, Project Management Institute, 2005.

[13] B. W. Boehm, "Software risk management: Principles and practices," IEEE Software, Vol.8, No.1, pp.32-41, 1991.

[14] J. Krebs, Agile Portfolio Management, Microsoft Press, 2009.

[15] V. Subramaniam and A. Hunt, Practices of an Agile Developer, The Pragmatic Programmers, LLC., 2006.

[16] K. Schwaber, SCRUM Development Process [Internet], <http://jeffsutherland.org/oops/schwabpub.pdf>

[17] K. Schwaber and M. Beedle, Agile Software Development with SCRUM, Pearson Education Inc., 2009.

[18] M. Poppendieck and T. Poppendieck, Implementing Lean Software Development: From Concept To Cash, Boston: Addison-Wesley Professional, 2007.



백 정 희

e-mail : junghee.baek@gmail.com
 1987년 아주대학교 전자계산학과(학사)
 1989년 아주대학교 컴퓨터공학과(석사)
 2011년 숭실대학교 미디어학과 박사과정
 수료
 관심분야 : 소셜앱매뉴팩처링, 모바일 컴퓨팅,
 클라우드서비스



임 영 환

e-mail : yhlim@ssu.ac.kr
 1979년 KAIST 전산학과(석사)
 1985년 Northwestern University 전산학과
 (박사)
 1996년~현 재 숭실대학교 미디어학과 교수
 관심분야 : 모바일 콘텐츠 및 솔루션, 멀티
 미디어