

민첩방법론을 활용한 오픈마켓 시스템 개발 프로세스 개선

이 용 희[†] · 이 건 호^{††}

요 약

본 연구에서는 민첩방법론을 활용하여 오픈마켓 개발 프로세스의 생산성 문제를 해결하고자 한다. 오픈마켓 기업을 대상으로 프로세스의 현황 분석과 문제점을 조사하고, 린 소프트웨어의 민첩 실천원칙을 활용하여 문제점을 해결한다. 또한 개선 효과를 입증하기 위해서 개선 전후의 프로세스 리드타임과 결함 발생을 비교 분석한다.

키워드 : 민첩방법론, 린 소프트웨어, 오픈마켓, 프로세스 개선

Development Process Improvement of Open-market System by using Agile Methodology

Yong-Hee Lee[†] · Gun-Ho Lee^{††}

ABSTRACT

This study improves the productivity of software development of open-market systems using agile methodology. The current processes of an open-market system are analyzed and inefficient processes are identified which are changed by applying the agile practice principles in Lean software. This study analyzes the process lead times and defects after the processes are changed to see the improvement effect.

Keywords : Agile Methodology, Lean Software, Open-Market, Process Improvement

1. 서 론

오픈마켓은 온라인상에서 개인 판매자가 직접 상품을 등록하여 매매가 이루어지는 시스템으로 유통 경로 단축과 판매자의 가격경쟁으로 소비자에게 다양한 상품을 저렴하게 제공하면서 빠른 성장을 보이고 있다[1]. 오픈마켓 성장과 함께 선 후발 업체간의 우위를 선점하기 위한 경쟁이 과열되면서, 신속한 서비스 개발이 매우 중요시되고 있다[2]. 그러나 오픈마켓이 대형화, 복잡화 되면서 납기 지연, 리소스 부족, 결함 발생으로 고객 신뢰 저해와 금전적 손해가 발생하고 있다. 이러한 문제를 해결하기 위한 다양한 연구가 진행되고 있으며, 최근 민첩방법론(Agile Methodology)이 관심의 대상이 되고 있다.

민첩방법론은 환경 변화에 신속하게 대응하기 위한 개발 방법론으로 2001년 민첩선언문(Agile Manifesto) 발표 후 본격

적으로 사용 되었다[3]. 주요 방법론은 XP(Extreme Programming), 스크럼(Scrum), 린 소프트웨어(Lean Software)가 있으며[4], 본 연구에서는 린 소프트웨어의 민첩 실천 원칙을 중심으로 프로세스를 개선하려 한다.

린 소프트웨어는 TPS(Toyota Production System)를 미국식 환경에 맞춰 재정의한 경영방법론인 린 시스템의 품질 기법을 소프트웨어 개발에 적용한 개발 방법론으로[5], 프로세스상의 낭비요소를 제거 후 결과를 측정하여 성과를 분석하게 된다. 린 소프트웨어에서 제시하는 낭비유형은 미완성 작업, 가외프로세스, 가외기능, 직무전환, 대기, 이동, 결함이 있으며[6], 이러한 낭비 요소를 제거하기 위한 7가지 원칙과 22가지 민첩 실천 도구를 제시한다[7].

린 소프트웨어는 2000년대 초반에 업계에 소개되면서 본격적으로 사용되었다. Babara는 워크플로우 관리방법 개선에 린 소프트웨어를 적용 한 사례를 소개하고, 활용 방안을 제시하였다[8]. Lehigh는 린 시스템과 6시그마를 활용하여 개발 프로세스를 개선하고, 결함 감소를 측정하여 효과를 검증하였다[9]. Raman은 린 소프트웨어를 CMM(Capability Maturity Model)과 소프트웨어 팩토리화 비교하여, 항공산

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

† 정 회 원 : 커머스플랫폼 개발관리팀 매니저

†† 종 신 회 원 : 숭실대학교 산업정보시스템공학과 조교수

논문접수 : 2010년 3월 22일

수정일 : 1차 2010년 3월 29일

심사완료 : 2010년 3월 29일

업 소프트웨어 개발에 적용 가능성을 확인하였다[10]. Perera는 기존 민첩 방법론과 린 소프트웨어를 결합하여 프로세스를 개선하고, 결함 감소를 측정하여 효과를 검증하였다[11]. 위와 같은 연구 동향에서 기존 연구는 활용 방안이 다소 구체적이지 못하고, 오픈마켓에 적용사례가 없는 단점이 있다.

본 연구의 목적은 린 소프트웨어를 활용한 민첩 방법론이 오픈마켓 개발 프로세스 개선에 유용함을 증명하는 것이다. 오픈마켓 기업을 대상으로 개발 프로세스를 분석하여, 발견된 문제점을 민첩 실천 원칙을 활용하여 해결하고, 프로세스 개선 전후의 프로세스 효율과 결함발생을 측정하여 개선의 효과를 검증하려 한다.

2. E기업의 개발프로세스 현황 및 문제점

2.1 E기업의 일반 정보 및 연구데이터

프로세스 개선 대상 기업은 한국 서울에 소재하는 오픈마켓 E기업이다. E기업의 조직은 사업부, 기획, UI(User Interface) 디자인, HTML 코딩, 개발, 테스트, 프로젝트 관리로 구성되며, 접수된 요구사항에 따라 각 팀에서 필요한 인력을 지정하여 프로젝트 팀을 구성하는 매트릭스구조로 운영되고 있다.

E기업의 프로젝트 현황과 결함분석을 위한 데이터는 2009년도에 수행한 114개의 프로젝트를 대상으로 하였으며, 결함 데이터 수집은 테스트 관리 툴인 Quality Center 10.0 [12]을 사용하였다.

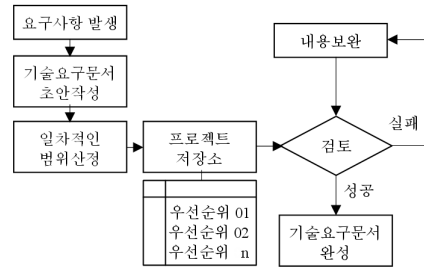
분석된 데이터 결과는 동시 진행 프로젝트 16개, 월평균 요청 건수 16개, 월평균 종료 건수 9개, 프로젝트당 월평균 소요 인력 4.85명, 월평균 대기 프로젝트 23개, 월평균 취소 프로젝트 1.8개, 프로젝트당 평균 결함 39.15개, 프로젝트당 평균 규모 106.85이며, 평균 프로젝트 규모(S)는 다음과 같은 식을 사용하여 계산한다.

- i: 프로젝트의 인덱스(i=1, ..., I)
- I: 전체 프로젝트 개수
- D_i: i번째 프로젝트 소요일자
- W_i: i번째 프로젝트 투입인력

$$S = \left(\sum_{i=1}^n W_i \right) / I \times \left(\sum_{i=1}^n D_i \right) / I$$

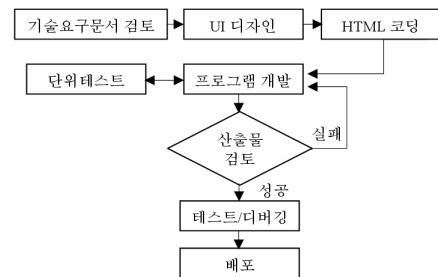
2.2 E기업의 개발 프로세스

E기업의 프로세스는 크게 두 부분으로 구분된다. 첫째는 요구사항 관리 프로세스로, 기획팀에서 사업부의 요구사항을 분석하여 기술요구문서 초안을 작성하여, 프로젝트저장소에 등록한다. 등록된 프로젝트는 부서별 리더가 참여한 회의에서 우선 순위를 결정하게 되며, 우선순위가 결정된 프로젝트 기술요구문서는 리뷰와 보완 절차를 거쳐 기술요구문서를 완성하게 된다. 다음 (그림 1)은 요구사항 관리 프로세스이다.



(그림1) 요구사항 관리 프로세스

둘째는 소프트웨어를 구현하는 프로세스이다. 프로젝트 참여자를 대상으로 요구사항을 설명하고 UI디자인, HTML 코딩, 개발, 테스트 단계를 거쳐 소프트웨어를 구현하게 된다. 소프트웨어 구현 프로세스는 다음 (그림 2)와 같다.



(그림2) 개발 프로세스

2.3 개발 프로세스의 문제점

E기업의 개발 프로세스의 개선점을 찾기 위하여 팀 별로 설문과 면담을 실시하였다. 조사된 문제점은 개발리드 타임이 길고, 커뮤니케이션의 오류가 많으며, 요구 사항이 과다하고, 진행 중 취소 되는 프로젝트가 많으며, 일부 대형 프로젝트와 개발자에서 결함이 편중되어 발생한다는 것이 있다.

3. 개발 프로세스 분석

3.1 개발 프로세스의 속도 측정

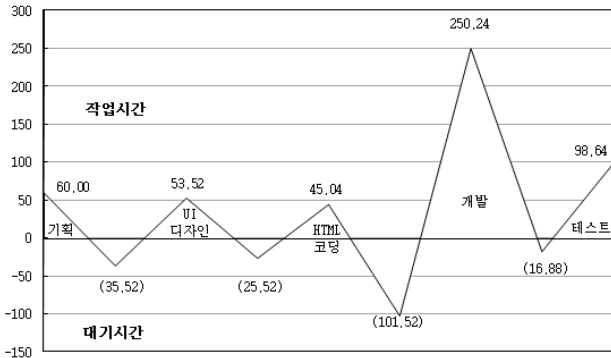
E기업의 프로세스 속도를 측정하기 위하여 대기시간과 작업시간을 측정하였다. 대기시간은 선행 공정이 종료되고, 다음 공정이 시작되기 전까지의 시간을 의미하며, 작업시간은 개별공정의 실행 시간을 의미한다. 다음 <표 1>은 측정된 프로세스 단계별 대기시간과 작업시간을 나타낸다.

다음 (그림 3)은 프로세스 단계별 대기시간과 작업시간에 대한 VSC(Value Stream Chart)이다.

측정결과 프로젝트를 완료하는데 평균적으로 22.43일을 대기하여 63.43일의 작업을 진행하였으며 프로젝트가 종료되기까지는 평균 85.86일이 소요되고 있으며, 프로세스의 효율은 73%를 나타낸다. 이때 대기시간이 긴 구간은 다음 단계의 작업이 원활하지 못해 병목현상이 발생하는 구간으로, HTML 코딩에서 개발로 넘어가는 구간에서 병목현상이 발생함을 확인 할 수 있다.

〈표 1〉 프로세스 단계별 대기시간과 작업시간

공정	대기	작업
기획	N/A	60.00
UI 디자인	35.52	53.52
HTML 코딩	25.52	45.04
개발	101.52	250.24
테스트	16.88	98.64
합계	179.44	507.44



(그림 3) 개선 전 VSC

3.2 개발팀의 인력 부족량 측정

개발단계에서 병목현상이 발생하는 주요 요인으로는 요구사항을 처리하기 위한 인력 부족이 있다. 개발단계에서 인력 부족량(W)은 다음과 같은 식으로 계산한다.

- P: 월평균 요청 프로젝트
- M: 프로젝트당 평균 소요인력
- R: 가용인력
- W = (P×M)-R

이에 대한 결과로 13.05 명의 개발인력이 부족함을 확인할 수 있다.

4. 개발 프로세스 개선

4.1 대형 프로젝트 결함 과다 발생

프로젝트 규모와 결함의 관계는 일반적으로 선형관계로 알려져 있다[13]. 하지만 대형 프로젝트에서 편향적으로 결함이 발생하는 것을 고려하였을 때, 규모와 결함발생의 관계가 곡선형이라고 가정하고 이를 증명하려 한다. 우선적으로 규모와 결함 두 변수의 관계의 여부를 확인하기 위하여 상관관계수 분석을 실시한다. 개발규모를 x, 결함발생을 y라고 했을 때 두 변수의 상관관계수(r)는 다음 식과 같다.

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

\bar{x}, \bar{y} : 각각 x, y의 평균값
 s_x, s_y : 각각 x, y의 표준편차

다음 <표 2>는 SPSS를 사용한 상관관계수 분석 결과이다.

〈표 2〉 상관관계수 분석 결과

변수	분석	결합	규모
결합	상관계수	1	0.878(**)
	유의수준. (1-tailed)	.	0.000
	N	82	82
규모	상관계수	0.878(**)	1
	유의수준. (1-tailed)	0.000	.
	N	82	82

분석 결과 두 변수의 상관관계수가 1로 유의하며, 0.878로 두 변수가 강한 관계가 있음을 확인할 수 있다. 회귀분석 적합도 평가를 통하여 Cubic 모형을 선택하였으며, 다음 <표 3>은 분석 결과에 대한 절편과 기울기를 나타낸다.

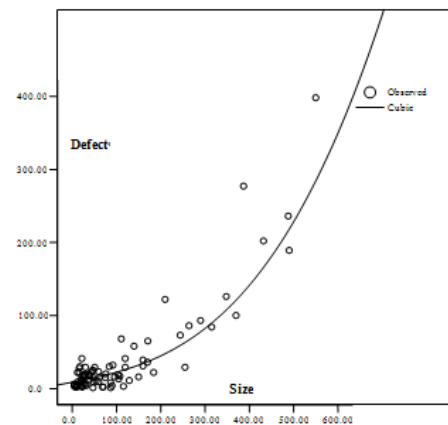
〈표 3〉 Cubic 회귀 분석 결과

변수	B	SE B	Beta	T	Sig T
규모	0.122	0.138	0.233	0.885	0.378
규모**2	0.000	0.000	0.129	0.200	0.842
규모**3	0.000	.0000	0.593	1.398	0.166
(Constant)	8.766	5.624		1.559	0.123

위 결과에서 도출된 Cubic모형의 결함 발생 방정식은 다음과 같다.

$$\text{결함} = 8.7668 + 0.1224\text{규모} + 0.0001\text{규모}^2 + 0.000001\text{규모}^3$$

결함 발생 방정식에 대입하면 규모 10~100 구간은 1.445개, 200~300 구간은 2.224개, 300~400 구간은 5.624의 결함이 발생한다. 다음 (그림 4)는 이러한 방정식을 그래프로 나타낸 것이다.



(그림4) 규모와 결함의 관계 그래프

프로젝트의 규모가 증가 할수록 결함의 발생은 기하급수적으로 증가 하게 되어, 대형 프로젝트는 규모를 분리하여 진행하는 것이 결함 발생을 감소시키게 된다.

4.2 불필요한 요구사항 과다

일반적으로 시스템의 기능 중 45%는 전혀 사용되지 않고, 19%는 거의 사용되지 않는다고 한다[14]. E기업에서는 불필요한 요구사항을 줄이기 위해서 프로젝트의 우선 순위를 정하고 있지만, 프로젝트를 요청하는 팀의 요구강도나 힘에 영향으로 우선순위가 결정되는 경우가 있다.

우선순위를 명확하게 하기 위하여 가치, 구현가능성 점수에 항목별 가중치를 적용한 매트릭스를 활용하고[15], 매출 성과를 재무 관점에서 평가할 수 있는 전문가를 선임하여 프로젝트의 우선순위를 조정하게 한다. 이 방법은 구현 가치가 높은 프로젝트를 우선 수행하게 하고, 불필요한 기능을 감소시킨다. 다음 <표 4>는 우선순위 매트릭스이다.

<표 4> 프로젝트 우선순위 매트릭스

우선 순위	프로젝트	프로젝트 가치						성공가능성					총합
		매출	순이익	고객만족	납기	비용절감	합계	기간	투자액	투입인력	난이도	합계	
	가중치	3	3	2	2	1		1	2	1	2		
1	PRJ06	4	3	3	5	2	39	3	1	2	1	21	60
2	PRJ02	2	5	3	4	4	39	2	3	2	4	12	51
3	PRJ05	5	1	3	4	3	35	5	2	3	2	14	49
4	PRJ09	2	3	4	3	4	33	1	5	4	2	11	44
5	PRJ07	3	4	2	1	1	28	3	2	2	3	15	43
6	PRJ10	3	4	2	2	5	34	2	4	2	5	8	42
7	PRJ03	3	2	2	3	3	28	3	3	5	2	12	40
8	PRJ01	3	4	1	1	5	30	1	5	5	4	6	36
9	PRJ08	1	2	3	2	3	22	2	3	3	4	11	33
10	PRJ04	1	2	1	4	2	21	5	4	4	3	7	28

매트릭스 상에서 프로젝트가치(V), 성공가능성(S), 총합(T)은 다음과 같은 식으로 계산한다.

P_{ki} : 프로젝트 k의 프로젝트가치 i번째 속성의 점수($i=1, \dots, 5$)

P_{kj} : 프로젝트 k의 성공가능성 j번째 속성의 점수($j=1, \dots, 5$)

W_i : 프로젝트가치의 i번째 속성의 가중치

Y_j : 성공가능성의 j번째 속성의 가중치

M: 속성의 최대 점수(5)

$$V_k = \sum_{i=1}^5 (P_{ki} \times W_i)$$

$$S_k = \sum_{j=1}^4 (M \times Y_j) - \sum_{j=1}^4 (P_{kj} \times Y_j)$$

$T_k = V_k + S_k$ (T_k 의 값이 클수록 우선 순위가 높은 것으로 결정)

4.3 프로젝트 취소

프로젝트가 진행 중 취소되는 경우 많은 낭비가 발생하

며, 취소 시기가 진행의 후반부로 갈수록 영향 범위가 크고, 진행된 내용을 원상복구하기 위한 추가비용이 발생하기도 한다. E기업의 경우 월평균 1.8개의 프로젝트가 진행 중 취소 되고 있으며, 이는 월평균 진행되는 프로젝트의 17.46%에 해당되며, 이에 투입되어 낭비되는 인력은 월 최대 8.73명이다.

프로젝트 취소를 줄이기 위해, 진행 전에 외부업체와의 협력이 필요하거나, 확정되지 않은 예산을 사용해야 하는 등 취소가 발생 할 수 있는 프로젝트에 대해서는 별도로 관리한다. 그 중에서 지연이 가능한 프로젝트와 제약사항이 정리되기 이전에 진행해야 할 프로젝트로 분류하고, 제약요소를 정리하여 가장 늦게 시작할 수 있는 시간을 파악하여, 취소 가능성이 있는 프로젝트의 시작을 늦추어 프로젝트 취소 시 발생하는 영향을 감소시킨다. 다음 <표 5>는 프로젝트의 제약사항을 정리한 예제이다.

<표 5> 프로젝트 제약사항 매트릭스

프로젝트	제약사항	제약만료일	사전작업일
수수료 변경	외주작업	2009-08-01	2009-07-25
	법률이슈	2009-08-25	N/A
검색결과 개선	예산확정	2009-08-11	N/A
	개발투입	2009-08-25	N/A
	외부디자인	2009-08-27	N/A

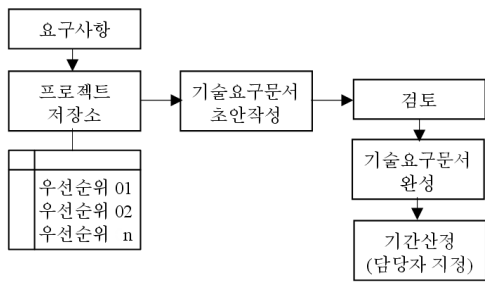
4.4 일부 개발자에게 결함 발생 편중

일부 개발자에게 결함 발생이 편중되어, 이를 분석결과 임시계약직과 신규 입사 인력을 위주로 발생함을 알 수 있었다. 이들에게서 결함이 편중되어 발생하는 원인을 스코핑(Scoping) 과정에서 발견하였다. E기업의 경우 프로젝트 투입 인력과 기간을 산정할 때, 평균적인 작업능력을 적용하여, 프로젝트 기간과 프로젝트에 투입될 인력 수를 지정하게 프로세스가 되어 있다.

개발자의 업무 능력은 최저와 최상에서 10배정도의 차이가 있을 수 있다[16]. E기업의 프로세스는 평균 업무 능력을 기준으로 스코핑하면서, 숙련된 개발자는 빠르게 업무를 처리 후 대기시간이 길어지게 되고, 비 숙련된 개발자는 작업 시간이 부족하게 되어 결과물에 대한 결함이 증가 된다.

이러한 문제를 해결하기 위해서 개발자의 개인 능력을 가장 잘 알고 있는 개발팀에 스코핑 과정을 위임한다. 개발팀에서는 개발기간을 산정하기 이전에 담당 개발자를 지정하여, 담당 개발자가 업무를 분석하고, 기간을 산정하여, 개발자의 능력차이를 스코핑에 반영하게 된다. 다음 (그림 5)는 개선된 스코핑 프로세스를 나타낸다.

개인별 작업능력차이를 고려한 스코핑으로 일부 개발자에게 결함 발생이 편중되던 현상이 감소하게 된다. 부가적인 효과로 개발 일정을 스스로 계획하여 책임감이 증가하고, 초기부터 업무 분석을 하게 됨으로 요구사항의 이해도가 높아지게 된다.

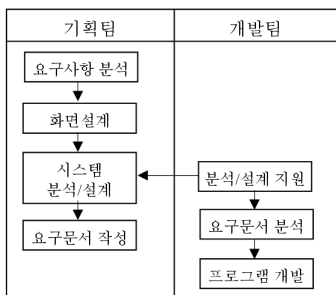


(그림 5) 개선된 스코핑 프로세스

4.5 커뮤니케이션 오류

프로젝트 진행 중 발생하는 커뮤니케이션 오류는 많은 리소스의 낭비를 가져온다. E기업은 기획팀에서 기술요구문서를 작성하여, 개발팀과 QA팀에게 전달한다. 이 과정에서 발생하는 커뮤니케이션 오류는 프로젝트 후반부에 다양한 낭비를 가져오게 된다. 본 연구에서는 문제의 원인을 기획자의 업무 범위에서 찾았다. 기획자는 요구 사항을 화면위주로 정리하고 있어, 전반적인 시스템의 이해가 부족하고, 요청사항으로 발생하는 영향을 고려하지 못한다.

이 문제를 해결 하기 위해 요구사항문서에 분석 및 설계 내용을 추가시킨다. 기획문서에 분석 및 설계를 추가하기 위해서는 해당 시스템에 대한 학습이 선행되어야 하고, 이를 작성하면서 시스템에 대한 이해가 높아지게 된다. 기획자의 시스템에 대한 이해가 높아지면 커뮤니케이션의 정확성과 속도가 높아져 오류가 감소하게 된다. 개발팀에서는 분석/설계 과정에 약 22%의 인력을 소요하고 있으며 기획팀에서 이 업무를 분업하게 되면, 개발팀의 업무 부하가 감소 된다. 다음 (그림 6)은 개선된 기획팀과 개발팀의 업무 프로세스를 나타낸 것이다.



(그림 6) 개선된 업무 프로세스

개선된 프로세스에서는 기획팀이 시스템 분석/설계 업무를 담당하고, 개발팀은 이에 대해서 리뷰 및 자료 제공 등에 대한 업무를 지원하게 된다. 결과적으로 기획자와 개발자는 요구문서가 완료되기 전부터 요구사항에 대한 논의가 이루어져 빠르고 정확한 커뮤니케이션이 가능하게 된다.

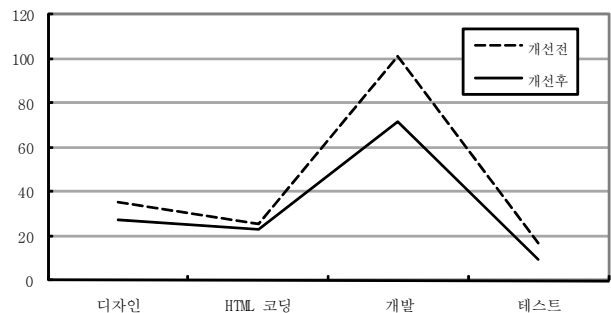
5. 프로세스 개선 결과 검증

프로세스 개선의 효과를 검증하기 위하여 개선 후 진행된 35개의 프로젝트를 대상으로 프로세스 대기시간과 결함 발생을 분석하였다. 다음 <표 6>은 개선 전후의 대기시간과 작업시간을 나타낸다.

<표 6> 개선 전후 대기시간 비교

공정	개선 전	개선 후	단축시간
UI 디자인	35.52	27.36	8.16
HTML 코딩	25.52	23.36	2.16
개발	101.52	71.92	29.60
테스트	16.88	9.60	7.28
합계	179.44	132.04	47.20

프로세스 개선 결과 개발단계의 대기시간이 29.50이 감소하여 병목 현상이 개선되었고, 전체 대기시간은 47.20이 단축 되었다. 다음 (그림 7)은 프로세스 개선 전후의 대기시간을 비교한 그래프이다.



(그림 7) 개선 전후의 대기시간 비교

프로젝트의 규모를 200이하로 조정하여 결함 발생을 측정 한 결과, 규모 1당 결함이 0.13감소하였다. 다음 <표 7>은 개선 전후의 결함 발생을 비교한 결과이다.

<표 7> 개선 전후의 결함 발생

항목	개선 전	개선 후
프로젝트 개수	114	35
평균 규모	106.85	52.66
평균 결함	39.15	12.51
규모 1당 결함	0.36	0.23

6. 결 론

본 연구에서는 오픈마켓 개발 프로세스 개선을 목적으로, 린 소프트웨어의 민첩 실천 원칙을 활용하였다. 프로세스를 분석하여 문제점을 찾은 후 낭비요소를 제거하여, 병목 구간이 되었던 개발단계의 효율을 높였으며, 개선 전후의 프로세스 시간과 결함 발생을 비교 분석하여 민첩방법론이 오픈마켓 프로세스 개선에 유용함을 증명하였다.

본 연구가 프로세스를 중심으로 개선하였으나 각 세부 공정이 개선되지 않으면 프로세스 개선은 한계를 가지게 된다. 추후 연구로는 다양한 민첩방법론을 활용하여 세부 공정에 대한 개선 연구가 진행되어야 한다.

참 고 문 헌

- [1] 신형원, "인터넷 쇼핑 시장의 변화와 대응전략," 삼성경제연구소, 2008.
- [2] 한광희, "우리나라 인터넷 쇼핑몰의 특징과 문제점 개선," 한국콘텐츠학회논문지, 제7권 제3호, pp.187-196, 2007.
- [3] Kent Beck, et. Al, "Manifesto for Agile Software Development", <http://agilemanifesto.org>, 2001.
- [4] Highsmith, J, Cockburn, A, "Agile software development: the business of innovation," IEEE Computer Society, Vol.34, No. 9, pp.120-122, Sept. 2001.
- [5] Wikipedia®, "Taiichi Ohno," http://en.wikipedia.org/wiki/Taiichi_Ohno, 2009.
- [6] Poppendieck, M, "Principles of Lean Thinking," <http://www.poppendieck.com/papers/LeanThinking.pdf>, 2002.
- [7] Mary Poppendieck & Tom Poppendieck, "Implementing Lean software development: from concept cash," Addison-Wesley, 2006.
- [8] Weber Barbara, Wild Werner, "Application of Lean and Agile principles to workflow management," Springer Berlin, pp.258-261, 2005.
- [9] Kevin Lehigh, "Improving software productivity using lean six sigma methods and tools," Crystal Ball User Conference, 2007.
- [10] Sowmyan Raman, "Lean software Development is it feasible?," Digital Avionics Systems Conference, Vol.1, pp. 1-8, Nov. 1998.
- [11] G.I.U.S. Perera, M.S.D. Fernando, "Enhanced Agile Software Development - Hybrid Paradigm with Lean Practice," ICIS, Aug. 2007.
- [12] Wikipedia®, "HP Quality Center," http://en.wikipedia.org/wiki/HP_Quality_Center, 2010.
- [13] Patton Ron, "Software Testing, 2nd Edition," SAMS, pp.306, 2006.
- [14] Jim Johnson, "ROI, It's Your Job," International Conference on Extreme Programming, pp.26-29, May. 2002.
- [15] Shankar Swaroop, "Managing Multiple Medium- and Small-scale Projects in Large IT Organizations," Information systems control journal, Vol.4, 2007.
- [16] Tom DeMarco, Timothy Lister, "Peopleware: Productive Projects and Teams," Dorset House, 1987.



이 용 희

e-mail : ta512k@naver.com

2008년 숭실대학교 산업정보시스템공학과 박사 수료

2000~2003 이모션 개발그룹

2003~2006 삼양데이터시스템 솔루션팀

2008~현 재 커머스플래닛 개발관리팀

관심분야: 프로세스개선, 프로젝트관리, 개발방법론



이 건 호

e-mail : ghlee@ssu.ac.kr

1996년 U. of Iowa, Dept of Industrial Eng.

1997~1999 숭실대학교 산업공학과 전임 강사

1999~2004 숭실대학교 산업정보시스템공학과 조교수

2004~현 재 숭실대학교 산업정보시스템공학과 부교수

관심분야: 소프트웨어공학, 데이터마케팅, 기계학습