

비즈니스 어플리케이션 운영관리 下에서 변경관리에 관한 연구

정승원*, 이석주**

*고려대학교 컴퓨터정보통신대학원

**고려대학교 컴퓨터정보통신대학원

e-mail : jsw94@korea.ac.kr*, seouklee@korea.ac.kr**

A Study on Change Management in the Evolution and Maintenance of Business Applications

Seung Won Jung*, SeoukJoo Lee**

*Graduate School of Computer Information & Communication, Korea University

** Graduate School of Computer Information & Communication, Korea University

요 약

급속한 기술환경의 변화와 정보화 사회의 도래에 따라 시스템의 구축 비용보다 더 많은 비용이 유지보수에 투입되고 있다. 소프트웨어의 생명주기 측면에서 40~90%의 큰 비중을 차지하는 유지보수를 효율적으로 관리하는 것은 매우 중요한 문제이다. 이에 본 연구에서는 비즈니스 어플리케이션의 운영관리에 관한 사례를 유지보수 유형에 따라 분석하고, 이를 바탕으로 변경 관리의 개선 방향을 제시하였다.

Keywords and Phrases : Software Evolution; Software Maintenance; Maintenance Categories; Project Change Management; Business Applications; Information Systems Operation; Software Support;

1. 서론

최근 급속하게 기술 환경이 변화하고 있고, 정보화 사회의 도래에 따라 IT 서비스는 기술 중심에서 프로세스 중심으로 패러다임이 이동하고 있다. 현재까지 소프트웨어 개발과 관련한 방법론 및 프로세스에 대한 연구는 꾸준히 이루어지고 있고, 지속적인 발전을 이루어 오고 있다. 하지만 제품 인도 후 시스템 유지보수에 대한 연구는 상대적으로 미미하다[1].

Keith(1998)의 연구에 의하면, 대부분의 소프트웨어 유지보수는 전체 생명주기 비용의 40~90% 가량을 차지하는 것으로 나타나고 있다. 미국과 영국의 많은 연구들은 소프트웨어 유지보수의 전체 비용을 계산해 내려는 노력을 해왔으며, 소프트웨어 유지보수에 있어서 많은 비용이 들어가는 것은 명백해 졌다[2]. 또한 소프트웨어 개발자 노력의 50% 이상이 유지보수에 사용된다[3]. 이와 같이, 전체 시스템 규모에서 매우 큰 비중을 차지하고, 많은 노력이 투입되는 유지보수를 효율적으로 관리하는 것은 매우 중요한 문제이다.

소프트웨어 엔지니어링 작업은 개발과 유지보수 두 부분으로 나누어 진다. 개발은 창의적이고, 흥미로운 영역으로, 반면에 유지보수는 판에 박히고, 번거로운 영역으로 구분된다. 이러한 이유로 소프트웨어

엔지니어링 이론은 수년 동안 유지보수를 무시하는 경향을 보였다. 하지만 유지보수 업무는 실무가 이론을 이끌어 오고 있는 실정이다[4].

본 연구의 목적은 비즈니스 어플리케이션의 운영관리 下에서 유지보수 속성을 Swanson(1976), Chapin(2001)의 유지보수 유형에 따라 분석하고, 변경관리에 관한 연구를 통해 개선 방향을 제시하고자 한다.

2. 문헌 및 선행연구

1) 유지보수의 정의 및 분류

소프트웨어가 개발되고 나면, 유지보수 단계로 진입한다. ANSI/IEEE Standard 610(1990)의 소프트웨어 유지보수의 정의는 아래와 같다[5].

소프트웨어 유지보수는 소프트웨어 시스템, 컴포넌트가 delivery된 후에 오류를 수정하고, 성능이나 다른 속성들을 향상시키거나, 환경 변화에 적응하기 위해 수정하는 프로세스이다.

Swanson(1976) 연구에서는 유지보수를 수정 유지보수(Corrective maintenance), 적응 유지보수(Adaptive maintenance), 완전 유지보수(Perfective maintenance)의 세 가지 유형으로 <표 1>과 같이 분류하고 있다[6].

<표 1> Swanson의 유지보수 카테고리 분류

구분	설명
수정 (Corrective)	프로세스 실패 성능 실패 구현 실패
적응 (Adaptive)	환경 변화 프로세스 변화
완전 (Perfective)	프로세스 비효율 성능 향상 유지 보수

Ned Chapin(2001) 연구에서는 소프트웨어 유지보수 유형을 4가지 클러스터, 12가지 유형으로 <표 2>과 같이 분류하고 있다[7].

<표 2> Chapin의 유지보수 카테고리 분류

구분		설명
클러스터	유형	
비즈니스 규칙	기능향상	비즈니스 규칙의 추가나 변경
	수정	설계 결함, 코딩 오류의 제거
	기능축소	사용자 기능의 축소나 제한
소프트웨어 속성	적응	다른 플랫폼/OS에서 작업
	성능	성능 향상
	예방	향후 발생할 유지보수를 줄임
	다듬기	소스코드를 다듬음.
문서화	내용수정	내용 수정(소스코드 관련 없음)
	포맷수정	포맷 수정(소스코드 관련 없음)
인터페이스 지원	평가	소프트웨어 평가(감사, 조사 등)
	컨설팅	SW가 협의, 자문 목적으로 사용
	훈련	SW가 훈련 목적으로 사용

2) 변경관리

ISO21500(2012):Guidance on project management에서는 변경 통제를 프로젝트와 인도물에 대한 변경을 통제하고, 변경을 수용하거나 기각하는 것을 공식화하기 위한 것으로 정의한다. 프로젝트가 수행되는 동안 변경 요청을 기록하고, 이익, 범위, 자원, 시간, 비용, 품질, 위험 관점에서 변경을 평가하고, 영향도를 분석하고, 실행에 있어 승인을 받는 절차가 필요하다. 변경이 승인될 때, 모든 이해관계자들에게 전달되어야 하며, 모든 인도물에 대한 변경은 형상관리와 같은 절차를 통해 통제되어야 한다[8].

프로젝트관리지식체계(PMBOK)의 통합 변경통제 수행은 모든 변경요청을 검토하고, 변경을 승인 하고, 인도물, 조직 프로세스 자산, 프로젝트 문서에 대한 변경사항을 관리하고, 변경사항 처리에 대한 내용을 전달하는 프로세스이다. 변경 사항을 검토, 평가, 승인, 보류 또는 기각할 책임과 결정사항에 대한 기록 및 의사소통을 담당 하기 위해 구성된 공인 위원회인 변경통제 위원회(CCB)가 통합 변경통제 수행 프로세스에 포함 된다[9].

3) 변경관리 사례 선행연구

황경태(2004) “정보시스템 운영 및 유지보수 업무에 관한 사례 연구”에서는 국내 대형

제조업체의 유지보수 업무를 기록한 사례를 분석하여 유지보수 업무 중에서 가장 비중을 많이 차지하는 변경요청 처리 업무의 현황을 best practice 와 비교하였다. 그 결과, 수정 유지보수 16.7%, 적응 유지보수 66.3%, 완전 유지보수 17.0% 로 수정 유지보수는 best practice 유사하나, 적응 유지보수의 비중이 매우 높게, 완전 유지보수의 비중은 매우 낮게 나타났다. 이러한 원인으로 시스템 개발시 사용자의 참여 미흡, 사용자 요구사항 파악 부족, 유지보수 인력의 개발업무 겸직 등을 들었다[10].

이문선(2010) “SI 프로젝트의 범위가 일정, 비용 및 리스크에 미치는 영향 분석 사례” 연구에서는 SI 프로젝트 241 개에서 종료 시 계획 대비 범위 변경현황, 구축 단계 말 일정 진척 현황 등을 분석하여, 범위 변경이 일정 및 비용 요인에 미치는 영향을 분석하였다. 분석결과 범위변경은 평균 10% 선에서 관리되고 있으며, 발주기관에 따른 차이는 없는 것으로 나타났다[11].

3. 연구방법 및 분석방법

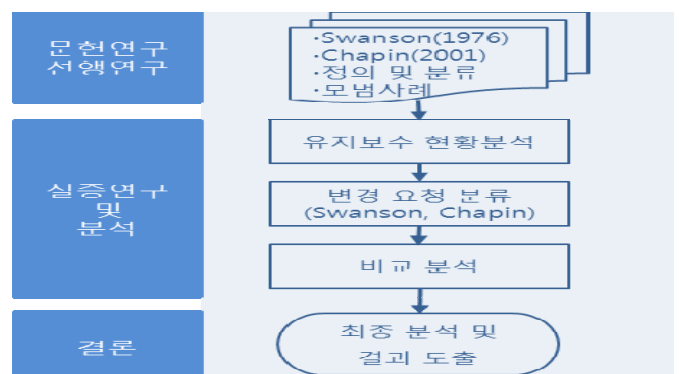
1) 연구방법

기존 문헌 및 선행 연구를 기반으로 Swanson(1976) 과 Chapin(2001)의 유지보수 유형 및 분류 방법에 대해 조사를 하였다.

국내 여러 기업의 시스템 통합 프로젝트가 종료된 이후, 유지보수 프로젝트에 착수 후 6 개월 동안 유지보수 데이터를 분석하였다. 지식관리 시스템, 위험 관리 시스템, 문서관리 시스템, 사전영업 관리 시스템 등 8개 시스템을 대상으로 한다.

유지보수 데이터 총 118 건중 변경기각, 변경 중지 건이 22 건, 변경완료 건이 96 건이며, 변경완료 건 중 Data 수정요청 19 건을 제외하고, 77 건의 변경요청에 관한 사항을 Swanson의 3가지 유지보수 유형과 Chapin의 12가지 유형으로 분류하였다. Chapin의 유형 중 인터페이스 지원 클러스터에 해당하는 평가, 컨설팅, 훈련의 경우 일반적인 변경 요청으로 분류가 어렵기 때문에 본 연구에서 이 부분은 제외한다.

이렇게 분류한 데이터와 변경속성 데이터를 비교 분석 함으로써, 최종 분석 결과를 제시하였다. 이에 대한 연구 접근 방법은 (그림 1)과 같다.



(그림 1) 연구접근 방법

2) 연구내용

Swanson 의 방식에서 완전 유지보수 하나로 분류 하던 것을 Chapin 의 방식에서는 기능향상, 기능축소, 성능, 예방, 다듬기, 내용수정, 포맷수정의 7 가지로 세분화 하고 있다. 또한 기존 Swanson 의 분류에는 없었던 평가, 컨설팅, 훈련과 같은 부분도 유지보수 유형에 새롭게 정의하였다. Chapin 의 연구에 의하면 Chapin 의 분류와 Swanson 의 분류는 다음 <표 3>와 같이 매핑된다[7].

<표 3> Chapin-Swanson 매핑

Chapin 분류		Swanson 분류
클러스터	유형	
비즈니스 규칙	기능향상	완전
	수정	수정
	기능축소	완전
소프트웨어 속성	적응	적응
	성능	완전
	예방	완전
	다듬기	완전
문서화	내용수정	완전
	포맷수정	완전
인터페이스 지원	평가	-
	컨설팅	-
	훈련	-

4. 분석 결과

1) 일반자료 현황

어플리케이션별 단순, 일반 현황은 <표 4> 와 같다.

<표 4> 어플리케이션별 단순-일반 분류 현황

구 분	단순 ¹⁾	일반 ²⁾	Total
A appl.	9	9	18
B appl.		7	7
C appl.		5	5
D appl.	1	1	2
E appl.	5	6	11
F appl.	15	9	24
G appl.		1	1
H appl.	6	3	9
Total	36	41	77

- 1) 단순 : 개발일정이 5MD 이하이고, 변경심의 활동 없음
- 2) 일반 : 개발일정이 6MD 이상이거나, 변경심의 활동 실시

단순 건은 전체 건수 대비 47%, 일반 건은 53%의 비율로 나타났으며, 주요 비즈니스 프로세스를 포함하는 A, F 어플리케이션에서 다수의 변경 건이 발생하였다.

단순, 일반 건별 평균 변경 처리 소요일은 <표 5> 와 같다. 단순, 일반 변경 건수는 유사하게 나타났으며, 평균 변경 처리일의 경우 일반 건들이 단순 건들에 비해 약 81% 정도 더 소요되는 것으로 나타났다.

<표 5> 단순-일반 평균 변경 처리 소요일

구 분	변경(건수)	평균 변경 처리일(M/D)
단순	36	7.1
일반	41	12.9
Total	77	-

2) 분석자료 현황

Chapin 의 유지보수 유형과 Swanson 의 유지보수 유형별로 구분하여 발생건수와 비중을 분석한 결과 <표 6>과 같다.

<표 6> Chapin-Swanson 의 분류에 따른 현황

구분	클러스터	유형	Swanson 분류			전체 건수	비중 (%)
			완전	수정	적응		
Chapin 분류	비즈니스 규칙	기능향상	58			58	75.3
		수정		10		10	13.0
		기능축소	4			4	5.2
	소프트웨어 속성	적응					
		성능					
		예방	3			3	3.9
문서화	다듬기	2			2	2.6	
	내용수정						
인터페이스 지원	포맷수정						
	평가						
	컨설팅						
전체 건수			67	10		77	
비중(%)			87.0	13.0			

본 분석 데이터 중 열 기준 분류인 Swanson 의 데이터를 살펴보면, 적응 유지보수가 한 건도 존재하지 않는다. 이것은 시스템 통합 프로젝트 종료 직후 이므로, 환경변화로 인한 변경 발생 가능성이 낮기 때문이다. 적응 유지보수 건이 없는 것을 고려할 때, 예러 수정인 수정 유지보수건의 비중은 13%로 비교적 낮은 편으로 나타났다.

본 분석 데이터 중 행 기준 분류인 Chapin 의 데이터를 살펴보면, 적응, 성능, 예방, 내용수정, 포맷수정 유형의 경우는 한 건도 존재하지 않으며, 전체 유지보수 중 가장 높은 비중을 차지하는 기능향상 유형은 75.3%에 이르는 것으로 분석되었다.

Chapin 의 분류 중 가장 높은 비중을 차지하고 있는 기능향상과 수정 건의 어플리케이션 별 현황은 <표 7>과 같다.

<표 7> Chapin 기능향상, 수정 현황

구 분	기능향상	수정
A appl.	16	2
B appl.	5	
C appl.	3	2
D appl.	2	
E appl.	7	2
F appl.	22	1
G appl.		1
H appl.	3	2
Total	58	10

주요 비즈니스 프로세스를 포함하고, 활발한 사용자를 가진 A, F 어플리케이션에서 다수의 기능향상건이 발생하였다.

3) 분석 결과

선행 연구자들에 의해 정의된 유지보수 유형에 따라 데이터를 분류하고 다각도로 분석한 결과는 아래와 같다.

첫째, 분석 데이터 중 변경 중지/기각율은 18.6%로써, 해외 선진 best practice 의 변경 기각율이 80%인 것에 비하면 변경 기각율이 낮은 편이다. 이는 변경요청 건이 이해관계자들에 의해 검토가 철저히 이루어지는지 검증해볼 필요가 있으며, 변경 요청에 대한 검토 절차를 강화할 필요가 있는 것으로 분석된다.

둘째, 비즈니스 규칙의 변화로 인한 변경 요청이 80.5%로 가장 큰 비중을 차지한다. 특히 기업의 주요 비즈니스 프로세스를 담고 있는 어플리케이션의 경우는 그 비중이 더 높다. 본 사례의 경우는 프로젝트 종료 직후의 데이터로서, 비즈니스 규칙(환경)의 변화에 의한 것 이라기보다는 사용자 요구사항의 변경으로 인한 영향이 더 큰 것으로 보인다. 이와 같은 현상이 발생하는 가장 큰 원인은 구축 프로젝트 과정에서 사용자의 참여 부족으로 사용자의 요구사항이 완벽하게 정의되지 못하였고, 프로젝트 팀원들의 협업 업무에 대한 이해 부족으로 사용자의 요구사항을 제대로 이해하지 못하였다고 볼 수 있다.

셋째, 적응, 성능, 예방, 내용수정, 포맷수정 유형의 경우는 한 건도 존재하지 않았다. 이는 구축 프로젝트에서의 통합 테스트, 성능 테스트 및 인수테스트가 철저하게 수행되고, 모든 결함 사항에 대해서는 프로젝트 종료 이전에 올바르게 수정이 되었으며, 이를 통해 운영 관리시 발생할 수 있는 변경 요청을 줄일 수 있었기 때문이라고 볼 수 있다.

이 결과를 통하여 비즈니스 어플리케이션의 운영관리 하에서 변경 요청 건을 줄일 수 있는 요소를 도출하였다.

5. 결론 및 향후 방향

비즈니스 어플리케이션에 대한 의존도가 증가하고, 복잡도가 높아짐에 따라 운영에 있어서 비용적인 측면과 전략적인 측면이 부각되고 있다. 그럼에도 불구하고 이에 대한 연구는 간과되어 왔다.

본 연구에서는 비즈니스 어플리케이션 운영관리 하에서의 변경 관리에 관한 연구를 통해 국내 기업의 유지보수 업무 현황을 분석하였고 이중, 변경요청을 유지보수 유형별로 분류하고 비교 분석함으로써 변경요청을 줄일 수 있는 요소를 도출해 보았다.

분석 결과를 정리하면 다음과 같다.

분석 데이터에서 변경 기각율은 약 18%이며, 이는 80%의 해외 선진 사례에 비해서 낮은 수준이다. 변경이 적절히 통제되지 않는다면, 소프트웨어의 복잡도를 증가시키고, 이를 운영하기 위해서 많은 비용이 들게 된다. 또한 분석 데이터에서 비즈니스 규칙의 변화로 인한 변경 요청은 전체의 80.5%로 주로 사용

자의 요구사항 변경에 의한 것으로 나타났으며, 프로젝트 진행시 사용자 요구사항 도출이 부족했기 때문이다. 사용자의 적극적인 참여를 유도하고, 사용자 중심적인 개발을 진행한다면 운영관리 하에서의 변경요청을 줄일 수 있을 것이다. 마지막으로 통합 테스트, 성능 테스트, 인수 테스트를 철저하게 수행한다면 운영관리 하에서 변경 요청을 줄이는 데 기여할 수 있을 것이다.

본 연구에서는 시스템 통합 프로젝트 종료 후 일정 기간 동안의 유지보수 데이터를 분석함으로써 인해 Chapin 의 일부 유지보수 유형은 나타나지 않았다. 유지보수 데이터 분석기간을 확대하고, 대상 어플리케이션 및 어플리케이션 속성을 확장하여 분석한다면 분석되지 않은 유지보수 유형을 포함하여, 더욱 의미 있는 데이터를 도출해 낼 수 있을 것이다.

참고문헌

- [1] 정정목, “공공기관의 정보시스템 유지보수에 영향을 미치는 요인에 관한 실증적 연구”, 『논문집(숭실대)』, 2015
- [2] Keith Bennett, “Software evolution: past, present and future” Information and Software Technology, Volume 38, Issue 11, November 1996, Pages 673-680
- [3] Bennett K.H., Rajlich V.T. (2000) Software maintenance and evolution: a roadmap. In: Proceedings of international conference on software engineering, Limerick, Ireland pp. 75-87
- [4] Glass, R.L., “The Relationship between Theory and Practice in software Engineering” Association for Computing Machinery, Communications of the ACM, Vol. 39, No. 11, November 1996, pp. 11-13.
- [5] IEEE Std. 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE, New York, 1991.
- [6] E. B. Swanson. The Dimensions of Maintenance. In ICSE '76: Proceedings of the 2nd international conference on Software engineering, pages 492{497, Los Alamitos, CA, USA, 1976. IEEE Computer Society Press
- [7] Ned Chapin. Types of software evolution and software maintenance. JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE 2001; 13:3-30
- [8] PMI(Project Management Institute) "A guide to the project management body of knowledge 5th edition"
- [9] ISO 21500:2012, *Guidance on Project Management*
- [10] 황경태, 남기찬, 김화식, “정보시스템 운영 및 유지보수 업무에 관한 사례 연구”, 한국정보기술응용학회, Vol. 11, Issue 3, p. 77-87, 2004
- [11] 이문선, 이석주, “SI프로젝트의 범위가 일정, 비용 및 리스크에 미치는 영향 분석” 『논문집(고려대)』, 2010