

효율적인 ITIL 변경관리 프로세스를 위한 개발기간 지표

마수정*, 김진형**

*고려대학교 컴퓨터정보통신대학원

**고려대학교 컴퓨터학과

e-mail : milker13@korea.ac.kr , koolmania@software.korea.ac.kr

Development Term Index for effective ITIL Change Management

Soojung Ma*, Jinhyung Kim**

*Graduate School of Computer&Information Technology, Korea University

**Dept. of Computer Science and Engineering, Korea University

요 약

기업의 IT 환경이 시스템 중심에서 비즈니스 중심으로 바뀌어감에 따라 IT 서비스 관리 프로세스 구축에 대한 가이드를 제공하는 Information Technology Infrastructure Library(ITIL) 모델이 나왔다. ITIL 적용의 가장 큰 목적은 IT 업무 프로세스 개선을 통한 서비스 품질 개선과 서비스 비용 절감이다. 변경관리 IT 환경의 모든 사고 및 변경에 관해 효율적이고 신속한 처리를 위한 표준화된 방법과 절차를 사용하는 ITIL의 Service Support 의 한 영역이다. 그러나 기존 모델에는 우선순위, 업무범위 등 처리절차에 대해 선진사례가 나와있는 반면 일정에 대한 신뢰성이 없다는 문제점이 있다. 따라서 본 논문에서 프로젝트 관리에서의 일정관리 개념을 도입하고 Goal Question Metrics(GQM) 기법을 사용하여 일정관리의 신뢰성을 부여할 수 있는 개발기간 지표를 제안한다. 또한 이러한 지표의 타당성 검증을 위해 S 사의 데이터를 비교 분석한다. 결과적으로 제안하는 지표를 통하여 조직운영의 효율성과 비즈니스에 초점을 맞춘 IT 서비스를 제공할 수 있다.

1. 서론

최근 비즈니스는 빠르게 변화하고 있다. 고객의 요구는 점차 까다로워지고 있으며, 기존 기업간의 경쟁이 격화되고 전혀 예측하지 못한 곳에서 새롭게 강력한 경쟁자의 출현이 빈번해지고 있다. 기업은 예측하지 못한 외부 환경의 변화에도 이를 빠르게 감지하고 탄력적이고 능동적으로 대응할 수 있는 능력이 요구된다. 이에 따라 기업들은 보다 효율적인 IT에 대한 요구가 높아지고 있으며 단일 기업에서 초차 IT 인프라를 관리하고 운영하는 정보시스템의 역할을 '서비스'관점으로 변경하기 위한 노력을 하고 있다.[1]

이를 위해 IT 조직에선 service desk, rule engine 등 신규개발과 더불어 기존 IT 조직원들의 일정관리를 통해 빠르게 대응하고자 한다. 일반적으로 유지보수 조

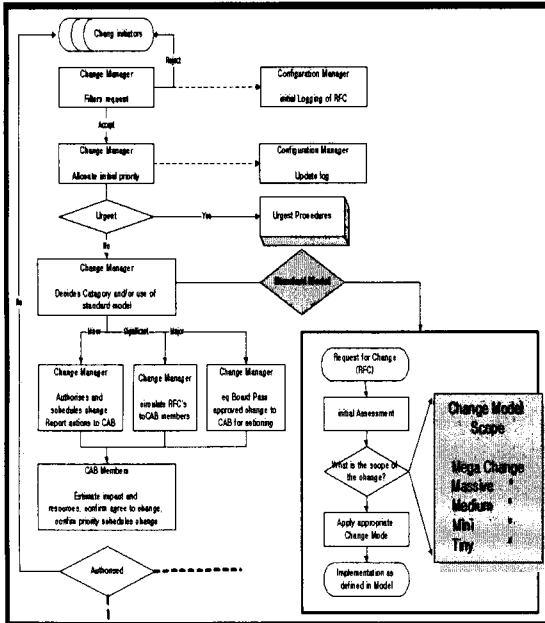
직에서는 업무의뢰가 오면 개발자의 경험적 판단에 의해 예상 완료 일을 산정하며 관리자는 예상 완료 일에 근거하여 업무실적을 평가한다. 이러한 과정은 예상 완료 일의 비무상적인 산정과 지연을 야기한다. 이는 변경관리 일정과 업무량 사이에 객관적인 지표가 없기 때문이다. 본 연구는 다음과 같은 내용으로 구성된다. 2 장에서는 관련연구로서 변경관리와 일정관리를 기술하며 두 관계를 측정하는 GQM 기법에 대해 소개하고 3 장에서는 일정관리의 신뢰성을 위한 개발기간 지표를 제안한다. 4 장에서는 기존의 업무처리 기법과 개발기간 지표를 적용한 업무처리 기법에 대한 비교평가를 수행하며 5 장에서는 결론과 향후 연구에 대해 상세 한다.

2. 관련연구

2.1 ITIL Change Management(변경관리)

ITIL 변경관리는 IT 환경아래의 모든 변경에 관해 효율적이고 신속한 처리를 위한 선진 사례 모음이다.[2]

[그림 1]은 변경관리 기본모델의 한 부분으로 업무 처리에 대한 우선순위, 투입인원, 업무범위 등의 처리 절차가 있다. 하지만 ITIL 변경관리에는 업무의 일정 관리에 대한 내용은 포함되어 있지 않다. 따라서 ITIL 변경관리를 통한 업무 처리의 경우 일정 관리 부분에서의 비효율성을 나타낸다. 이에 개발지표의 활용을 통해 CM Scope 에 예상기간을 주어 프로세스를 개선 시키고자 한다.



(그림 1)변경관리 프로세스

2.2 PM Time Management (일정관리)

프로젝트 관리란 프로젝트 요구사항을 충족시키는 데 필요한 지식, 기량, 도구 및 기법 등을 프로젝트 활동에 적용하는 것을 말한다. 프로젝트관리는 프로젝트 착수, 기획, 실행, 감시, 통제 및 종료 단계로 진행되는 프로젝트관리 프로세스의 적용과 통합을 통해 이루어진다.[11]

PM 에서는 Time Management 를 이용하여 업무일정을 관리한다. 일정관리에는 활동정의, 활동순서, 기간산정, 일정개발, 일정통제라는 중요개념이 포함되며, 각 개념의 세부항목은 다음의 <표 1>과 같다.

2.3 목표기반 GQM (Goal Question Metrics)기법

변경관리와 프로젝트관리의 개념을 접목시키는 방법으로는 GQM 기법을 사용한다.[6] GQM 은 목적, 질문, 측정 3 단계로 목적에 적합한 측정치를 선택하는 체계적인 방법이다. GQM 접근방법은 프로세스 품질개선을 위한 단순한 형태로서, NASA 프로젝트에서 결함을 평가하기 위해 개발된 것이다. 이 접근 방법은 조

직이 그 목적을 수립하기 위해서는 의미 있는 척도 (meaningful measure)의 도입과 데이터 수집을 위한 운영적 기반을 제공하는 데이터와 연결된다. GQM 은 다음과 같은 3 수준으로 이루어진다.[6]

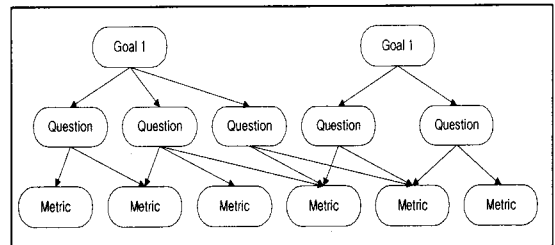
<표 1> 일정관리

일정 관리	투입물	도구와 기법	산출물
활동 정의	1 작업 분류 체계 2 범위 명세서 3 과거기록 정보 4 재약사항 5 가정 6 전문가 판단	1 분할 2 모형(templates)	1 활동 목록 2 보조 세부사항 3 작업 분류 체계 개선
활동 순서	1 활동 목록 2 결과물 설명 3 필수 선후관계 4 임의 선후관계 5 외부 선후관계 6 마일스톤	1 선후형도형법(PDM) 2 화살도식법(ADM) 3 조건도식법 4 네트워크 모형	1 프로젝트 네트워크도 2 활동 목록 개선
활동 기간 산정	1 활동 목록 2 재약사항 3 가정 4 자원 요구사항 5 자원 수용능력 6 과거기록 정보 7 식별된 위험	1 전문가 판단 2 유사 산정 3 계량 기간 4 후보일정(예비)	1 활동 기간 산정치 2 산정 근거 3 활동 목록 개선
일정 개발	1 프로젝트 네트워크도 2 활동 기간 산정치 3 자원 필요 사항 4 자원 저장 설명 5 달력 6 재약사항 7.가정 8 선행 및 지연 9 위험 관리 계획 10 활동 속성	1 수학적 분석 2 기간 단축 3 시뮬레이션 4 자원배정Heuristics 5 PM 소프트웨어 6 코드 체계	1 프로젝트 일정 2 보조 세부사항 3 일정 관리 계획 4 자원 필요사항 개선
일정 통제	1 프로젝트 일정 2 진척 보고서 3 변경 요구 4 일정 관리 계획	1 일정 변경 통제 시스템 2 성과 측정 3 부가 계획 4 PM 소프트웨어 5 차이 분석	1 일정 개선 2 일정 조치 3 교훈

-목적(Goal) : 개념적 수준으로 개선활동에서 연관된 대상의 범위를 정의한다.

-질문(Question) : 운영적 수준으로 특정 목적의 심사/달성이 수행되는 방법을 특성화하는 질문을 사용한다.

-측정(Metric) : 정량적 수준으로 각 질문에 연관된 데이터를 연관시켜 주관적, 객관적 데이터를 제공한다.



(그림 2) GQM 방법

GQM 방법은 목적에 적합한 메트릭을 선택하는 체계적인 방법이다. 또한 필요로 하는 메트릭에 집중할 수 있게 해주기 때문에 데이터 수집에 드는 비용을 줄여준다. 또한 GQM 방법의 적용에서 나오는 메트릭은 소프트웨어 제품, 프로세스, 개발과정의 특성을 정량화한다.[2]

3. 개발기간지표의개발

3.1 중요요소 추출

GQM 방식으로 PM 일정관리 주요항목에 ITIL 변경관리 내용을 접목해 변경관리 중요요소를 추출한다.

1 단계는 개념적인 목표인 PM의 주요개념이다
2 단계는 핵심성공요소인 PM의 주요개념에 속하는 ITIL의 요소관점에서 합당한 질문을 한다.[4]

- IT 조직에서 일어나는 변경관리에 해당되는 업무분류에는 어떤 것들이 있는가?
 - 인시던트 또는 문제보고서에 요구된 해결방법
 - 고객접촉이나 SLA를 통해 나타난 사용자나 고객 의 불만
 - 제안된 새로운 CI의 도입 또는 제어
 - 인프라스트럭처의 몇몇 구성요소에 대해 제안된 업그레이드
 - 변경된 비즈니스 요구 지시
 - 새롭거나 변경된 법률
 - 위치변경
 - 벤더 또는 계약자로부터 제품 또는 서비스 변경
- 비즈니스적으로 문제와 그 해결을 위한 영향에 기반하여 우선순위를 정하는가?
 - 낮음(변경은 합리적이고 필수적이지만, 다음에 계획된 릴리즈 또는 업그레이드 때까지 기다릴 수 있다. 적절하게 자원할당)
 - 중간(심각하지 않은 영향, 그러나 다음에 계획된 릴리즈 또는 업그레이드 때까지 수정을 미룰 수는 없다. 자원에 대한 중간 우선순위)
 - 높음(심각하게 영향 받는 몇몇 사용자들 또는 다수의 사용자들에 영향,변경구성,테스트,실행자원에 대해 가장 높은 우선순위)
 - 즉시(다수의사용자들,중대한 역할의 시스템, 또는 몇몇 심각한 문제에서 서비스 손실 또는 위험한 문제 발생 이것은 즉각적인 행동이 요구되며, 긴급 CAB 또는 CAM/EC 모임의 소집이 필요하다. 또한 그러한 권한이 주어진 변경의 구성 시 즉시 자원이 할당되어야 함)

- 변경되거나 신규 작성되는 모듈의 수는 몇 명인가?
 - 투입인원의 수는 몇 명인가?
 - 모듈 사이의 상호연관성은 통계적으로 어떠한가?
 - 가용성 및 신뢰성을 위해 해결방법은 무엇인가?
 - 문서작업 및 품질관리에 얼마큼의 노력이 드는가?
- 3 단계에는 측정값 <표 2>을 제시한다.

<표 2> 중요요소 추출 매트릭스

1 단계 (PM)	2 단계 (CI)	3 단계 (Values)
활동정의	업무분류	재작업 : 1 인시던트 해결 : 2 비즈니스 변경 : 3 지속적인 불만 : 4 시스템 개발 : 5

활동순서	우선순위 변경의수	측시:1 높음:2 중간:3 낮음:4
활동기간	투입규모	혼자:1 CAB 구성: N CAB/EC 구성 : N+1
	변경영향	없음:0 데이터:0.6 제어:0.7 외부:0.8 공통:0.9 내용:1
일정통제	품질관리	변경깊이 재작업:1 임시방편:0.5 근본해결:1 Q

3.2 개발기간(DT)결정

개발기간을 추정하기 위해 Kitchenham[5]은 21 개의 개발일정 GQM 방식으로 PM 일정관리 모델을 제안하였다. 그러나 이 모델은 1990년대 이전에 개발된 프로젝트 대상으로 하고 있어 현대의 복잡하고 다양한 환경에서 개발되는 프로젝트들에 이들 모델들을 적용하기 위한 정확성과 가치를 판단할 수가 없다. 본 논문에서는 개발기간 모델을 결정하기 위한 판단기준으로 GQM 방법을 통해 나타난 중요요소 중에 비즈니스적 측면이 있는 투입규모(N), 우선순위(P), 업무분류(T)를 선택하여 계획 수립 시에 목표기간을 설정하는 개발기간이라 하였다.[2]

$$DT = N * (P+T)$$

3.3 개발기간 지표와 모델수준

지표란 상태가 가늠이 되는 표시 또는 특징이다.[3] 개발모델에 대한 지표의 설정을 위해 중요요소와 개발기간간의 관계를 설정하였다.

<표 3> 개발기간 지표와 모델수준

DT	개발기간지표	개발모델
1 ~ 2	3 시간	Tiny CM
3 ~ 4	8 시간	Mini CM
5 ~ 6	15 시간	Medium CM
7 ~ 8	25 시간	Massive CM
9 ~	40 시간	Mega CM

4. 검증

본 논문의 제안을 검증하기 위해 S사의 유지보수 업무수행 데이터를 대상으로 적용하였다. 연구의 타당성 및 결과의 신뢰성을 주기 위해 간단히 FP(Function Point)사용하였다. 진행의 개요는 다음과 같다. [7][9]

- Step1 : 지난 8개월간의 기능성과 복잡성을 고려한 100건에 대해 업무분류와 우선순위를 결정한다.
- Step2 : 프로그램 반출부터 이행 사이의 시간을 개발시간에 대한 데이터들의 평균값을 취한다.
- Step3 : 신규 발생되는 업무에 대해 Step1의 분포와 같도록 만든다.
- Step4 : Step2와 같은 작업을 통해 결과값에 대한 정확성을 비교한다



(그림 3) 데이터 추출화면

<표 3>에서 보는 바와 같이 개발기간지표를 사용한 개발기간의 단축을 의미하지는 않는다. 그러나 몇 가지 일반적인 현상이 나타났다.[5][8]

<표 4> 정확성확인

중요요소		지표설정 전			지표설정 후			
투입 인원	업무 분류	우선 순위	예상 시간	개발 시간	차이	지표	개발 시간	차이
1	1	1	8	2	6	2	2	1
		2	8	4	4	3	5	3
		3	8	7	1	4	7	1
		4	8	10	2	5	12	3
	2	1	8	5	3	3	5	3
		2	8	10	2	4	10	2
		3	16	12	4	5	13	2
		4	16	17	1	6	15	0
	3	1	16	7	9	4	8	0
		2	16	13	3	5	16	1
		3	24	18	6	6	16	1
		4	24	26	2	7	23	2
	4	1	24	12	2	5	14	1
		2	32	15	7	6	15	0
		3	32	30	8	7	27	2
		4	40	30	10	8	30	5
5	1	40	18	12	6	16	1	
	2	40	32	8	7	23	2	
	3	40	43	3	8	27	2	
	4	40	41	1	9	42	2	

- 결론 1 : 기존평균예상시간 대비 기존평균 개발시간의 차이에 비해 개발기간지표에 따른 예상시간 대비 실제평균 개발시간의 차이가 적다.
- 결론 2 : 노력을 통해 의해 객관적 지표에 의한 개발기간에 맞추므로써 성과측정의 판단기준으로 활용될 수 있다.
- 결론 3 : 지표에 의거한 모델은 조직관리자에게 예상기간에 대한 의사결정 정보를 제공한다.

5. 결론

소프트웨어 개발기간을 추정하기 위해 기존 연구들은 개발노력이 개발기간에 가장 큰 영향을 미치는 요인이라 추정하였다.[9] 그러나 비즈니스 요인을 기반한 객관적인 지표에 의해 보다 큰 영향을 받을 수 있다. 따라서 본 논문에서는 업무분류와 우선순위에 기반한 개발기간 지표를 제안하고 정형화된 유지보수 조직에서의 데이터를 비교분석 함으로써 타당성을 입증하였다. 신뢰성 있는 개발기간 지표를 기초해 관리자는 업무분배의 근거가 생겨 조직운영의 효율성을 갖게 되었으며 예상 완료 일 안에 업무를 수행함으로써 성과측정의 판단 기준을 만들었다. 그러나 소프트웨어 규모에 따라 최소한의 기간은 개개인의 숙련도에 많은 영향을 받을 수 있으나 본 논문은 이 부분을 고려하지 못하였다. 추후 이 분야에 대한 연구 및 업무분배, 성과측정에 대한 검증을 수행 할 것이다.

참고문헌

- [1] Rich Schiesser, ITSM IT System Management, 2005
- [2] 박찬순, “가중치기반의 애플리케이션 사용성 측정 방법”, 한국정보처리학회, 제 13 권, 제 1 호, 2006.05
- [3] 한혁수, 김한샘, “소프트웨어 개발 프로젝트 제어를 위한 재작업 지표의 적용”, 한국정보처리학회. 제 13 권, 제 1 호. 2006.02
- [4] ITIL, Service Support, Service Delivery
- [5] Kitchenham, B.A.. “Empirical Studies of Assumptions That Underlie Software Cost-estimation Model”. Information and Software Technology, Vol.34. No4, pp211~218,1992
- [6] V.R Basili, G Caldiera, H.D.Rombach. “The Goal Question Metric Approach”,1994
- [7] Daniel M.German “ Measuring fine-graines change in software : towards modification-aware change metrics” 11th IEEE international Software Mtrics Symposium,2005
- [8] 권기태/신수정/박찬규. “기능점수와 소프트웨어 측정”.그린.2003
- [9] 박석구 이상균. “기능점수 기반 소프트웨어 개발팀 규모와 개발기간 예측 모델”, 한국정보처리학회, 제 10-D 권, 제 7 호, 2003.12
- [10] Wolfhart Goethert and Will Hayes, “Experiences in Implementing Measurement Programs”, Technical Note CMU/SEI-2001-TN-026, Software Engineering Measurement and Analysis Initiative,2001
- [11] PMBOK Guide-Third Edition Exposure Draft