

프로젝트 환경과 특성에 따른 소프트웨어 개발 경로 선정 알고리즘

정 병 권[†] · 윤 석 민^{††}

요 약

본 논문은 프로젝트 환경과 특성을 고려하여, 소프트웨어 개발 경로를 커스터마이징 하는 방법을 설계한 것이다. 개발 경로 선정 기준은 ISO/IEC TR 15721 Information Technology Guide for ISO/IEC 12207(Software Life Cycle Processes), ISO/IEC 15504 Information technology - Process assessment의 프로세스 내용에 기반 한 10개의 영역 특성 이다. 10개의 영역 특성은 프로젝트 환경과 특성을 반영한 것이며, 프로젝트 개발 경로 선정 항목들을 정의한다. 프로젝트 환경과 특성 항목인 10개 영역 특성 항목을 이용하여 소프트웨어 개발 프로세스를 선정하는 알고리즘을 제시한다.

본 논문에서 제안한 소프트웨어 개발 경로를 커스터마이징 하는 방법을 평가하기 위하여 소프트웨어 개발 경로를 커스터마이징 하는 방법을 반영한 개발 경로 선정 시스템을 가지고 웹 기반 시스템 개발 프로젝트를 적용한다. 그러나 입증된 수작업 프로젝트 경로 프로세스와 시스템에서 출력된 프로젝트 경로 프로세스가 차이가 난다. 차이가 나는 주 원인은 산출물이 서로 합쳐졌거나, 명칭이 바뀌었기 때문이다. 효과로는 소프트웨어 개발 프로젝트 환경과 특성에 적합한 경로 프로세스를 용이하게 선정한다.

키워드 : 프로젝트 환경과 특성, 소프트웨어 개발 경로, 소프트웨어생명주기, 소프트웨어 유형, 프로젝트 규모, 프로젝트 난이도

Algorithm selecting Software development route suitable for Project environment and characteristics

Byung-Kwon Jung[†] · Seok-Min Yoon^{††}

ABSTRACT

This paper focused on the method for customizing software development path, considering the project environments and characteristics. The selection standard of development path is composed of ten items, based on the process of ISO/IEC TR 15721 Information Technology Guide for ISO/IEC 12207 (Software Life Cycle Process) and ISO/IEC 15504 Information technology - Process assessment. The ten items were reflected the project environments and characteristics, at the same time the items conduct the adjustment item of selecting project development path. An algorithm for selecting software development path through items of the project environments and characteristics is presented.

To test the algorithm in this paper, a system for selecting development path, which reflected algorithm, was developed. The development project for web-based system were also adopted to the system for selecting development path. In addition, provened hand-worked project path process differed from machine-worked project path process. The reason why it differs is that outputs were mixed or their names were changed. The effect is to select easily software development route suitable for project environment and characteristics.

Key Words : Project Environment and Characteristics, Software Development Path, Software Life Cycle, Software Types, Project Size, Project Complexity

1. 서 론

보통 대규모의 소프트웨어 프로젝트에 수반되는 작업 프로세스들이나 관련된 작업 자원들은 보통 매우 큰 복잡성을

지니며, 이를 관리하는 것은 대부분 전문가 개인들의 지식과 경험에 의존하여 왔다. 따라서 그러한 관리 과정은 매우 많은 시간이 소모되며 잦은 오류가 발생된다. 결과적으로 이러한 전문가 의존적인 방식은 소프트웨어 개발비용을 높이는 원인이 되며 개발된 소프트웨어의 품질도 보장 받기 어려워진다. 또한 소프트웨어 개발 프로젝트의 규모에 따라 적합하게 적용하지 않고, 소규모의 프로젝트도 소프트웨어

[†]정 회 원 : 두원공과대학 테크노경영계열 교수
^{††}정 회 원 : 동양공업전문대학 소프트웨어정보과 교수
논문접수 : 2005년 8월 12일, 심사완료 : 2005년 12월 6일

개발 방법론에 나와있는 모든 내용을 수행하려다 보면 많은 개발비용이 소요되어 방법론 적용에 문제가 되곤 한다[1].

이러한 문제점들을 해결하기 위한 방법으로 소프트웨어 프로세스와 관련 지식들을 재활용하는 접근 방법들이 제시되어왔다[2, 3]. 그러한 관리 과정은 매우 많은 시간이 소요되며 잦은 오류가 발생하게 된다. 이러한 방법들은 주로 재활용 정보저장소의 내용과 구조, 저장 및 검색 방법들에 연구의 초점을 맞추어 왔다. 그러나 이러한 연구들은 새로운 프로젝트를 위한 소프트웨어 프로세스를 조정해 내는데 한계점이 있다. 프로세스 생성 및 적용을 위한 프로토타입 시스템들은 주로 한정된 영역에서의 특정 프로젝트를 위한 프로세스를 생성하는데 초점을 맞추어 왔고, 일반적인 다양한 영역을 위한 프로세스 생성은 지원하지 못한다.

본 논문에서는 프로젝트의 환경과 특성 요인들을 분석하여 그들에 맞는 프로세스들을 선정하는 방법을 제시한다. 프로젝트 환경과 특성¹⁾에 맞는 소프트웨어 개발 경로²⁾ 선정용 용이하게 하기 위하여 ISO/IEC TR 15271[4], ISO/IEC 15504[5]의 프로세스 내용에 준하여 10개 영역 특성을 분류하여[1] 프로젝트 개발 경로 선정 항목을 정하여 이를 본 논문에서의 적용범위로 한다.

소프트웨어 개발 경로를 커스터마이징하는 방법을 적용하고 평가하기 위하여 소프트웨어 개발 경로를 커스터마이징하는 방법이 반영된 개발 경로 선정 시스템을 이용하여, 기존에 입증된 프로젝트의 수작업 경로 프로세스와 개발 경로 선정 시스템의 결과로 나온 경로 프로세스를 비교분석 한다. 소프트웨어 개발 경로를 커스터마이징하는 방법이 반영된 개발 경로 선정 시스템을 국제표준 ISO/IEC 15504의 평가 방법을 이용하여 평가한다. 그리고 본 논문의 개선 내용을 기술한다.

2. 관련 연구

소프트웨어 개발 프로젝트 경로 프로세스를 선정하는 방법으로 프로젝트의 과거자료를 이용함으로써 유사한 프로젝트 수행 시 과거 프로젝트 이력을 참조하여 과거의 시행착오를 줄이는 방안이 있다. FODA(Feature-Oriented Domain Analysis) 및 ODM(Organization Domain Modeling) 도메인 분석 기법에서는 재사용 가능한 요소들을 인덱싱하고 검색하는 방식을 제시한다. 그러나 이러한 연구는 도메인의 특성으로부터 소프트웨어 프로세스를 추출할 수 있도록 해주었으나 다양한 응용 도메인에 적용하기에는 한계점을 가지고 있다 [6].

1) 프로젝트 환경과 특성: ISO IEC TR 15271 5.3. Identify the project environment and characteristics항에 정책, 조직목표, 문제영역, 조직문화, 요구 사항, 프로젝트생명주기, 기술위험, 안정성, 사용기술, 특정프로세스의 성숙도에 의하여 정하여진다고 정의되어 있으며 본 논문에서는 이를 기반으로 <표 1> 영역특성으로 정함.
 2) 소프트웨어 개발 경로: ISO/IEC 12207에서 제시한 소웨어 개발 생명주기 프로세스를 반영한 소프트웨어 개발 방법론에서 소프트웨어 개발에 사용되는 작업과 산출물들을 말하며 ISO15504에서는 프로세스 평가하는데 사용되는 process와 work product들을 말한다. 본 논문에서는 <표 2>에서 나타난 소프트웨어 개발에 사용되는 방법론의 작업과 산출물들을 말한다.

프로젝트의 과거 경험을 재활용하기 위한 정보 저장소에 관한 연구로는 PAL(Process Asset Library)과 같은 대표적인 연구들도 있으며 그 프로토타입도 제시되었다[7, 8]. 조직 학습분야의 연구에서는[2, 9] 과거 프로젝트 중 현재 프로젝트 특성과 유사한 프로젝트들로부터 유용한 경험 지식을 추출할 수 있는 방법을 제시한다. 이러한 모형들은 정보저장소 구조, 소프트웨어 프로세스 모델링 및 그 사용방법에 초점을 맞추고 있어, 저장된 사례를 새로운 프로젝트 환경에 적용시키는 메커니즘을 제시하지 못하고 있다.

CS/10000 시스템은 특별히 클라이언트 서버 환경의 프로젝트를 위한 시스템으로 프로젝트 특성에 맞는 하드웨어와 소프트웨어 제품 그리고 프로토타입 생명주기 모형 중심의 소프트웨어 개발 프로세스를 선정하는데 반하여 본 논문의 개발 경로 선정 기준은 프로젝트 환경과 특성을 정의한 ISO/IEC TR 15721 Information Technology Guide for ISO/IEC 12207(Software Life Cycle Processes), ISO/IEC 15504 Information technology-Process assessment의 프로세스 내용을 기반으로 한 10개의 영역 특성의 선정 항목들로 생명주기모형, 소프트웨어 개발 방법론, 프로젝트 규모, 난이도에 적합한 작업과 산출물을 선정한다.

3. 소프트웨어 개발 경로 선정 알고리즘

본 논문의 범위에 해당되는 ISO/IEC TR 15721 Information Technology Guide for ISO/IEC 12207(Software Life Cycle Processes), ISO/IEC 15504 Information technology-Process assessment에 근거한 소프트웨어 개발 프로젝트 환경과 특성에 준하여 소프트웨어 개발 프로세스 선정에 사용되는 영역 특성은 <표 1>과 같다. <표 1>의 영역 특성에 근거하여 프로젝트 환경과 특성에 맞는 소프트웨어 개발 경로를 선정하기 위한 절차 흐름도는 (그림 1)과 같다. (그림 1)을 보면 첫번째 절차로 프로젝트 개요에 대한 내용은 프로젝트 번호, 프로젝트 명, 프로젝트 관리자, 프로젝트 후원자, 프로젝트 목적 및 범위, 시작일, 종료일, 예상 투입인원, 기술위험 목록, 정책 목록, 주요 성공요소를 등록한다. 프로젝트 개요는 향후 프로젝트 경로 프로세스를 조정할 때 필요하다. 예로 위험을 해결하거나, 정책을 반영하기 위하여 경로 프로세스를 조정 시킬 수 있다. (그림 1)의 각 절차의 설명을 보면 <표 1>의 영역 특성과 관련한다.

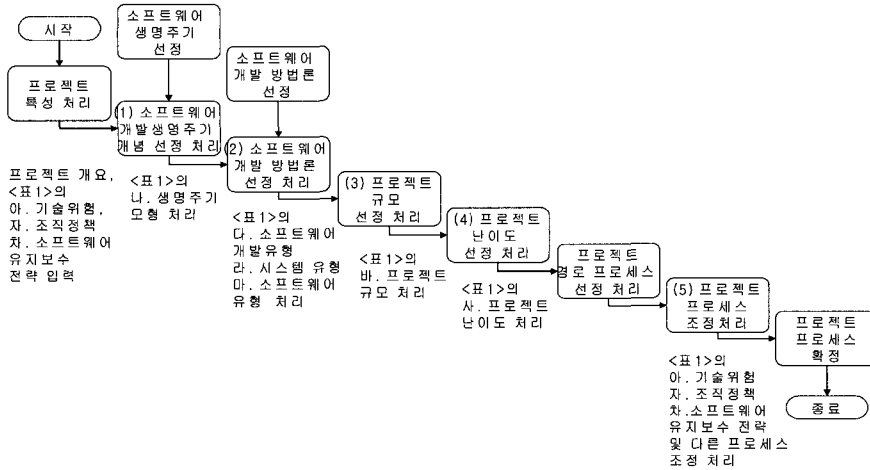
(그림 1)의 절차들은 본 단원의 소단원에서 설명한다.

소프트웨어 개발 경로를 선정하기 위하여 프로젝트 개요 자료사전, 소프트웨어 개발 방법론 작업절차 자료사전과 작업별 산출물 자료사전에 프로젝트의 특성을 저장하고 있다. 소프트웨어 개발 구성요소의 하나인 작업별 산출물 자료사전은 <표 2>와 같다.

<표 2>를 보면 여러 작업에서 동일한 산출물을 작업하면 산출물 코드는 다르다. 클라이언트 서버, 웹, 데이터웨어하우스, 패키지, 폭포수 모형, 프로토타입 모형, 증분 모형, 위험 관리 모형, 진화인도 모형, RAD 모형, 객체지향, 컴포넌트에

<표 1> ISO/IEC TR 15721, ISO/IEC 15504에 근거한 소프트웨어 개발 프로세스 선정에 사용되는 영역 특성

영역 특성	선택 유형	항 목
가. 시스템 생명 주기 단계		엔지니어링/개발
나. 개발생명주기 개념 모형	단일선택	① 폭포수 모형, ② 프로토타입 모형, ③ 진화적 인도 모형, ④ 중분, ⑤ 나선형, ⑥ RAD
다. 소프트웨어 개발 유형	단일선택	① 신규개발, ② 패키지 수정, ③ 프로토타입
라. 시스템 유형	단일선택	① 인터넷과 인트라넷 시스템, ② 단독 시스템, ③ 데이터웨어 하우스 시스템
마. 소프트웨어 유형	단일선택	① 윈도우 단독, ② 웹 기반, ③ 웹 기반과 데이터베이스, ④ 인트라넷
바. 프로젝트 규모	단일선택	①소, ②중, ③ 대
사. 프로젝트 난이도	단일선택	① 낮음, ② 중간, ③ 높음
아. 기술 위험	다중선택	① 미숙한 기술, ② 복잡한 시스템 인터페이스, ③ 기존의 HW, SW 제약, ④ 개발방법 부족, ⑤ 케이스 도구와 저장소 필요, ⑥ 성능 요구사항, ⑦ 재사용 라이브러리 존재, ⑧ 기존 시스템과의 인터페이스, ⑨ 내 고장성 (fault tolerance) 요구사항, ⑩ 객체지향 설계와 프로그래밍(컴포넌트 포함)
자. 조직 정책	다중선택	① 보안 정책, ② 안전 정책, ③ 위험 관리, ④ 독립적인 확인과 검증 작업, ⑤ 아웃소싱, ⑥ ISO 9000 인증
차. 소프트웨어 유지보수 전략	다중선택	① 장기간 큰 변화 ② 교육



(그림 1) 소프트웨어 개발 경로 선정 절차 흐름도

<표 2> 방법론 작업-산출물 자료사전

데이터사전명	한/영 용 도	산출물/d_output				작 성 일 작 성 자 내 용	2005.3.31 정 병 권 비 고
		소프트웨어 개발 산출물 테이블					
순번	항목명(한글)	항목명(영문)	형태	자리수	누계		
1	산출물코드	d_code	X	10	10	산출물코드	PK
2	산출물명	d_name	X	30	40	산출물명	
3	작업코드	d_tcode	X	10	50	작업코드	
4	작업명	d_tname	X	30	80	작업명	
5	영역	d_area	X	8	88	산출물작성요령 경로	
6	산출물작성요령경로	d_pic_path	X	15	103	산출물작성요령 경로	
7	산출물양식경로	d_pic2_path	X	15	118	산출물확인경로	
8	클라이언트서버	d_cs	X	1	119	클라이언트 시스템 개발	Y/N
9	웹	d_web	X	1	120	웹시스템개발	Y/N
10	데이터웨어하우스	d_dw	X	1	121	데이터웨어하우스개발	Y/N
11	폭포수	d_water	X	1	122	폭포수모형절차	Y/N
12	프로토타입	d_prototype	X	1	123	프로토타입모형절차	I/T/P
13	패키지	d_package	X	1	124	패키지이용개발	Y/N
14	중분	d_incremental	X	1	125	중분관리	Y/N
15	위험	d_risk	X	1	126	위험관리	Y/N
16	진화인도주문	d_order	X	1	127	진화인도주문	Y/N
17	진화인도비주문	d_stack	X	1	128	진화인도비주문	Y/N
18	RAD	d_rad	X	1	129	RAD 모형절차	Y/N
19	객체지향	d_object	X	1	130	객체지향	Y/N
20	컴포넌트	d_component	X	1	131	컴포넌트	Y/N
21	소규모	d_ssize	X	1	132	소규모선택	Y/O/N
22	중규모	d_msize	X	1	133	중규모선택	Y/O/N
23	대규모	d_lsize	X	1	134	대규모선택	Y/O/N
24	최초작성일자	t_startdate	D	8	142	최초작성일자	
25	최종변경일자	t_enddate	D	8	150	최종변경일자	

〈표 3〉 소프트웨어 생명주기 개념 선정표

평가요소	평가	단계접근			시제폭화			증분접근			진화접근			위험관리			전환접근		
		기준	가중	평점	기준	가중	평점	기준	가중	평점	기준	가중	평점	기준	가중	평점	기준	가중	평점
요구사항	명확	1.5			불명확	1.5		보통	1.0		보통	1.0		보통	1.0		명확	1.5	
개발기간	충분	1.5			보통	1.0		보통	1.0		부족	1.0		충분	1.0		부족	1.5	
분할가능	곤란	1.0			보통	1.0		용이	2.0		보통	1.0		곤란	1.0		용이	1.0	
개발도구	보통	1.0			구비	2.0		보통	1.0		보통	1.0		구비	1.0		부족	1.0	
개발경험	충분	1.5			충분	1.5		보통	1.5		충분	2.0		충분	1.5		보통	1.0	
개발인력	충분	1.5			보통	1.0		부족	1.5		보통	1.5		보통	1.0		부족	1.5	
위험도	보통	1.0			보통	1.0		보통	1.0		높음	1.0		높음	2.0		보통	1.0	
규모	중형	1.0			중형	1.0		대형	1.0		대형	1.5		대형	1.5		소형	1.5	
합계																			

관련된 산출물인지를 구분하기 위하여 항목을 두었으며, 프로젝트의 규모에 따른 선정여부를 정하기 위하여 항목을 두었다. 항목을 이용하여 경로 작업과 산출물을 선정한다.

3.1 소프트웨어 개발 생명 주기 개념 선정

소프트웨어 개발 생명 주기 개념을 선정하기 위하여 <표 3>의 소프트웨어 생명주기 개념 선정표를 활용한다[10].

평가요소의 평가항목에 3단계의 기준(명확, 보통, 불명확 등)을 기록하여 각 생명주기의 해당기준 선택은 3점을 주고 1구간 차에 ±1점을 적용하여 점수×가중치를 합한 결과를 가지고 가장 큰 점수의 생명주기를 선정하는 것이다.

생명주기 개념이 선정되면 <표 4> 폭포수 모형 소프트웨어 개발 프로세스와 프로젝트 관리 프로세스를 이용한 소프트웨어 개발 생명주기 모형 적용 방안을 따르며[11] 이를 <표 5>의 생명 주기 모형 선정에 따른 소프트웨어 개발 프로세스 선정 알고리즘에 기술한다.

<표 1>의 나. ① 폭포수 모형 선정의 경우 <표 2>의 방법론 작업-산출물 자료사전에서 d_water 항목이 “Y”인 작업과 산출물을 선정하면 된다. 나. ② 프로토타입 모형을

선정할 경우 인터페이스 프로토타입, 정보기술 프로토타입, 프로토타입 프로토타입을 선정하게 되면 <표 4> 폭포수 모형 소프트웨어 개발 프로세스와 프로젝트 관리 프로세스를 이용한 소프트웨어 개발 생명주기 모형 적용 방안과 같이 해당되는 작업과 산출물이 선정한다. 한가지의 경

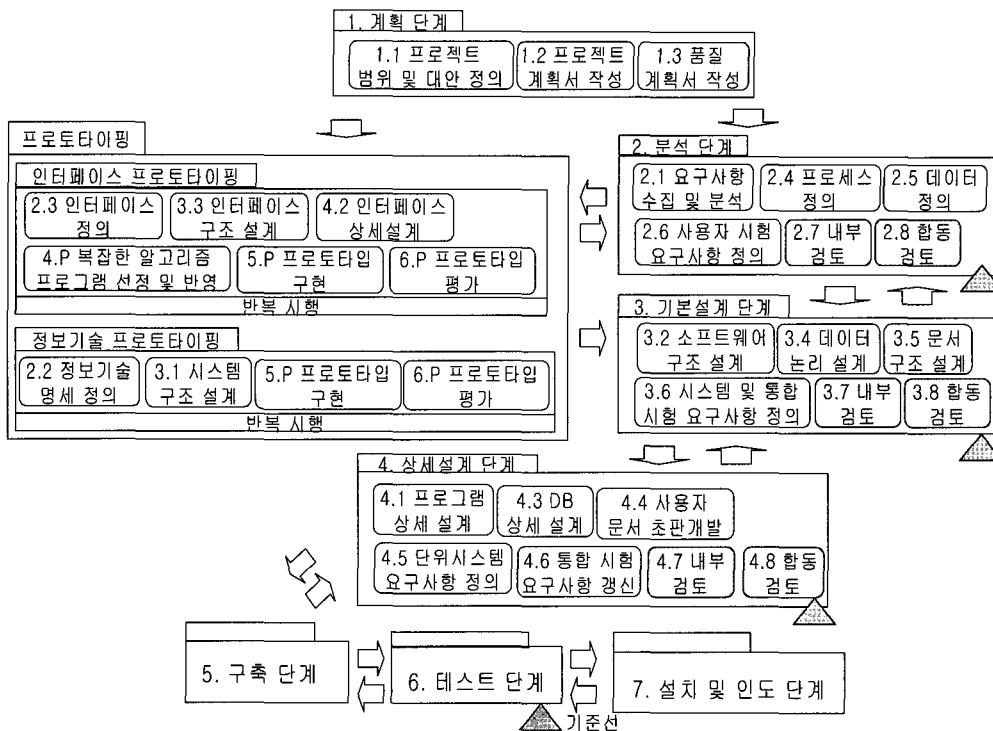
〈표 4〉 소프트웨어 개발 생명주기 모형 적용 방안

개발 생명 주기 모형	구분	개발프로세스	관리프로세스	처리내용
단계적 접근 모형		○		폭포수 모형 선정
프로토타입 모형	인터페이스 프로토타입	○		인터페이스 프로토타입 선정(I)
	정보기술 프로토타입	○		정보기술 프로토타입 선정(IT)
	파일럿 프로토타입	○		파일럿 프로토타입 선정(P)
진화인도모형	단기간	○		파일럿 프로토타입과 동일
	유지보수	○	○	파일럿프로토타입+Version관리 파일럿프로토타입+α,β 테스트
증분 모형	분석단계이후 증분		○	업무 분할 후 단계적 접근 모형 또는 프로토타입 모형 선택
	설계단계이후 증분		○	
	단계별 증분		○	
위험관리모형			○	프로젝트 관리의 위험관리지식 영역을 단계적 접근모형 또는 프로토타입 모형의 단계별 시행
RAD		○	○	파일럿프로토타입+속연인원선정+자동화도구선정+속연된 프로젝트 관리

〈표 5〉 생명 주기 모형 선정에 따른 소프트웨어 개발 프로세스 선정 알고리즘

```

DO CASE
CASE Water_Fall : 폭포수 모형 선정(<표1>의 나. ① 지원)
  Select_Model_Code = "W" : 선정된 절차 순서적으로 처리
CASE Prototype : 프로토타입 모형 선정 (<표1>의 나. ② 지원)
  Select_Model_Code = "P" : 선정된 절차 프로토타입 절차 분리하여 표현
CASE Evolutionary : 인터페이스 관련된 절차 프로토타입 절차로 분리
  Select_Model_Code = "F"
CASE IT : 정보기술 프로토타입 절차 프로토타입 절차로 분리
  Select_Model_Code = "T"
CASE Prototyping : 인터페이스 + 정보기술 프로토타입 절차 프로토타입 절차로 분리
  Select_Model_Code = "P" : 프로토타입 선정 테스트.(<표1>의 나. ④지원)
CASE Evolutionary : 진화적 인도 모형 선정 (<표1>의 나. ③ 지원)
DO CASE
CASE Short_Term : 단기간 진화 프로젝트
  Select_Model_Code = "P": 프로토타입 프로토타입과 동일
CASE Long_Term : 유지보수 프로젝트
DO CASE
CASE Order_Made : 주문 생산의 경우
  Select_Model_Code = "O": 프로토타입 절차조정+ Version관리 절차 추가
CASE Stock_Made : 비주문 생산의 경우
  Select_Model_Code = "M": 프로토타입 절차 조정 + α테스트절차와 β테스트 절차 선정
ENDCASE
ENDCASE
CASE Increment : 증분 모형 선정(<표1>의 나. ④ 지원)
  Select_Pra_Model_Code = "I" : 프로젝트 분할 시점 선정 및 프로젝트 분할 절차 선정
Perform Softer_Life_Cycle_Select_Model : 생명주기 선정 알고리즘 재수행
CASE Spiral : 나선형 모형 선정(<표1>의 나. ⑤ 지원)
  Select_Pra_Model_Code = "S" : 프로젝트 위험관리 절차 선정
Perform Softer_Life_Cycle_Select_Model : 생명주기 선정 알고리즘 재수행
CASE RAD : 고속개발 모형(<표1>의 나. ⑥ 지원)
  Select_Model_Code = "R" : 프로토타입 프로토타입 + 속연인원선정절차+도구선정절차+도구 활용절차, 효율적관리를 위한 절차 선정
ENDCASE
ENDCASE
  
```



(그림 2) 방법론 공통 활동의 프로토타이핑 모형

우를 설명하면 인터페이스 프로토타이핑의 경우 $d_{water} = "Y"$ 인 경우와 $d_{prototype} = "I"$ 인 경우를 선택한다. 그리고 (그림 2)에서와 같이 프로토타이핑 활동을 나누는 경우는 선정된 작업과 산출물에서 $d_{prototype} = "I"$ 인 경우 인터페이스 프로토타이핑 활동으로 별도로 관리하고 나머지는 일반 활동이다. 특히 4P 복잡한 알고리즘 프로그램 선정 및 반영, 5P 프로토타입 구현, 6P 프로토타입 평가 절차는 프로토타입 모형 선정인 경우에 선택하는 절차이다. 프로토타이핑 모형이 선정되면 프로토타이핑 관련 작업 산출물과 다른 일반 작업 산출물들은 동시에 수행할 수 있도록 (그림 2)와 같이 활동을 분리하여 관리한다[11].

<표 1>의 나. ③ 진화적 인도 모형의 경우 단기간의 경우 파일럿 프로토타이핑 모형 경로와 동일하며[12, 13], 장기간의 경우 유지보수와 같으며 주문 생산 소프트웨어 개발 경우 파일럿 프로토타이핑 모형 경로에서 선정된 작업 산출물과 일정한 유지보수기간이 지나면 수행하여야 하는 버전 관리의 작업과 산출물이 선정한다[14]. 버전관리 작업과 산출물 선정은 $d_{order} = "Y"$ 이다. 비주문 소프트웨어 개발 생산의 경우 $d_{stock} = "Y"$ 인 작업과 산출물을 선정하며, 이것은 α 테스트와 β 테스트인 작업과 산출물이 선정된다. 나. ④ 증분 모형 선정의 경우 어느 시점에서 개발할 어플리케이션을 분할하여 프로젝트 범위를 확정할 것인지 정하여 그 시점에 프로젝트 수행단위를 결정하기 위하여 $d_{increment} = "Y"$ 인 작업과 산출물을 선정한다. 이것은 업무범위를 분할하는 작업과 산출물이다. 그리고 다시 생명주기 선정 알고리즘을 수행한다. 나. ⑤ 나선형 모형의 경우 프로젝트의 위험관리 계획 작업과 산출물을 각 단계의 단계준비 활동에

서 $d_{risk} = "Y"$ 인 작업과 산출물을 선정토록 하고 단계검토 활동에서 위험 평가 작업과 산출물이 선정된다. 그리고 다시 생명주기개념 선정 알고리즘을 수행한다. 나. ⑥ RAD 모형의 경우는 파일럿 프로토타이핑 모형의 작업과 산출물을 선정하고 단계준비 절차에 $d_{rad} = "Y"$ 에 의하여 RAD를 위한 도구선정, 전문가선정, 도구활용 방법론 검토, 효과적인 프로젝트관리방안 작업과 산출물을 선정한다[15].

3.2 소프트웨어 개발 방법론 선정

소프트웨어 개발 방법론 선정 의사 결정표는 <표 6>과 같다. 나. 소프트웨어 개발 유형, 라. 시스템 유형, 마. 소프트웨어 유형의 프로젝트 특성 별로 CS, Web, DW를 지원하는 정보공학, 구조적 방법론, 객체지향 방법론, 컴포넌트 방법론을 선택한다.

시스템 유형이 단독사용인 경우는 웹 기반, 웹 기반과 데이터베이스 소프트웨어 유형은 관련이 없다. 방법론 선정에 따른 소프트웨어 개발 프로세스 선정 알고리즘은 <표 7>과 같다.

<표 6> 소프트웨어개발방법론 선정 의사결정표

방법론종류	정보공학, 구조적		객체지향	컴포넌트
프로젝트 특성 다. SW 개발 유형	① 신규개발 ② 패키지 수정 ③ 프로토타이핑			
라. 시스템 유형	① 인터넷/ 인트라넷	② 단독 사용	③ DW	모두가능 모두가능
마. SW 유형	① 원도우단독	CS	DW	객체지향 컴포넌트
	② 웹기반	Web	Web+DW	객체지향 컴포넌트
	③ 웹기반+DB	Web	Web+DW	객체지향 컴포넌트
	④ 인트라넷	CS	CS+DW	객체지향 컴포넌트

〈표 7〉 방법론 선정에 따른 소프트웨어 개발 프로세스 선정 알고리즘

```

DO CASE
CASE Information_Structural : 정보공학+구조적 방법론 (<표1, 6>의 다.①, ②, ③ 지원
                        : <표 1>의 나. ②인 경우 다. ③조건이 필수임)
DO CASE
CASE CS
Select d_cs = "Y" : 클라이언트+서버 시스템 개발 절차 선정 코드
                  : (<표1, 6>의 라. ① ② 경우 마. ①, ④ 지원)
CASE Web
Select d_web = "Y" : 웹 기반 개발 처리절차 선정 코드
                  : (<표1, 6>의 라. ① ③ 경우 마. ② ③ 지원)
CASE DW
Select d_dw = "Y" : 데이터웨어하우스 개발 처리 절차 선정 코드
                  : (<표1, 6>의 라. ③ 경우 마 지원)
END CASE
CASE Object_Component : 객체지향+컴포넌트 방법론
                      : <표1, 6>의 다. 지원
                      : <표 1>의 나. ②인 경우 다. ③조건이 필수임)
DO CASE
CASE Object
Select d_object = "Y" : 객체지향 절차 선정 코드
                    : (<표1, 6>의 라. 지원. 마. 지원)
CASE Component
Select d_component = "Y" : 컴포넌트 절차 선정 코드
                       : (<표1, 6>의 라. 지원. 마. 지원)
END CASE
ENDCASE
    
```

하나의 경우를 설명하면 CASE Information_Structural ; 정보공학 · 구조적 방법론은 <표 6>의 다. 소프트웨어 개발 유형의 ① 신규개발 ② 패키지 수정과 ③ 프로토타이핑을 지원한다. 그러나 나. 생명주기 모형에서 프로토타이핑 모형이 선정되면 프로토타이핑을 지원한다. CASE CS의 경우는 <표 2> 방법론 산출물 자료사전에 클라이언트 · 서버 항목 d_cs 에 "Y"가 표기되어야 한다. 이는 <표 1> 라. ① 인터넷과 인트라넷 시스템과 라. ② 단독 시스템인 경우, 마. ① 윈도우 단독사용 기반 소프트웨어와 마. ④ 인트라넷 기반 소프트웨어를 지원한다.

3.3 프로젝트 규모 선정

프로젝트 규모 선정에 따른 소프트웨어 개발 프로세스 선정방안은 <표 2>의 자료사전에서 보면 프로젝트 규모 항목이 소규모, 중규모, 대규모로 구분되어 작업과 산출물을 필수 선택으로 선정할 수 있도록 한다. 이를 지원하기 위하여 정보처리회사의 프로젝트 규모별로 산출물을 선정한 도표는 <표 8>과 같다.

〈표 8〉 프로젝트 규모에 따른 산출물 선정 지침

단계	단계 No	문서명	작성주관	1,2개월 소규모		3-6개월 이내 중규모		9개월 이상 대규모	
				선 택	필 수	선 택	필 수	선 택	필 수
개발 계획	1	프로젝트 개요서	감/용			1	1		
	2	비즈니스 모델 정의서	감	1	1		1		
	3	웹 사업요구 정의서	감	1	1		1		
	4	웹 요구기능 목록	감	1	1		1		
	5	업무배경도(context diagram)	감/용	1	1		1		
	6	기능분해도	용/감	1	1		1		
	7	주요엔티티 연관도	용/감				1	1	
	8	웹 주요 엔티티 목록 (주요엔티티 리스트)	용/감				1	1	
	9	원인 및 동기 리스트	감				1	1	
	10	웹 사이트 개요서 (응용시스템 개요서)	용/감				1	1	
	11	데이터 개요서	용/감				1	1	
	12	정보기술 개요서(정보시스템 개요서)	용/감				1	1	
	13	시스템 사양서	용/감				1	1	
	14	정보관리 개요서	용/감				1	1	
	15	프로젝트 산출물 내역서	용/감	1	1		1	1	
	16	개발계획서(프로젝트 계획서)	용/감	1	1		1	1	
	17	보고계획서	용/감				1	1	
	18	단계점검결과서	용/감				1	1	

〈표 9〉 프로젝트 난이도에 따른 규모 선정 알고리즘

```

Decide_Project_Size_as_Complexity(p_complexity,
p_size, p_con_size)
DO CASE
CASE p_complexity = "High"
DO CASE
CASE p_size = "Large"
p_con_size = "Large"
CASE Size = "Middle"
p_con_size = "Large"
CASE p_size = "Small"
p_con_size = "Middle"
ENDCASE
CASE p_complexity = "Middle"
p_con_size = p_size
CASE p_complexity = "Law"
DO CASE
CASE p_size = "Large"
p_con_size = "Middle"
CASE p_size = "Middle"
p_con_size = "Small"
CASE p_size = "Small"
p_con_size = "Small"
ENDCASE
CASE p_complexity = "Non"
p_con_size = "Non"
ENDCASE
    
```

<표 8>에 의해서 소프트웨어 개발 방법론 자료저장소에 프로젝트 규모표시를 할 수 있다. 이는 소프트웨어 난이도와 연계하여 <표 9> 프로젝트 난이도에 따른 규모 선정 알고리즘에 반영한다.

3.4 소프트웨어 난이도 선정 처리

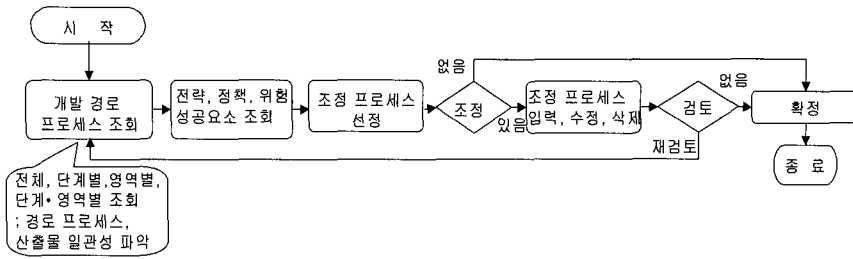
<표 1> 사. 프로젝트 난이도는 <표 9> 프로젝트 난이도에 따른 규모 선정 알고리즘과 같다. 난이도를 방법론의 관점영역인 전략, 업무, 프로세스, 객체, 컴포넌트, 데이터, 인터페이스, 정보기술, 품질관리, 프로젝트 관리별로 선정케 하여 난이도가 높은 해당 영역에 관련된 소프트웨어 개발 작업과 산출물 선정 방법을 프로젝트 규모를 한 단계 높여 선정하고 그리고 프로젝트 난이도가 낮으면 난이도가 낮은 해당영역에 관련된 소프트웨어 개발 작업과 산출물 선정 방법을 프로젝트 규모를 한 단계 낮게 또는 관련 없음을 선정하여 소프트웨어 개발 산출물을 정한다. 소프트웨어 난이도 선정처리 알고리즘은 <표 10>과 같다.

3.5 프로젝트 경로 프로세스 선정 처리

프로젝트 경로 프로세스 선정 처리는 <표 5>, <표 7>, <표 9>, <표 10>의 알고리즘을 동시에 만족하는 작업과 산출물을 선정하는 것이다. 소프트웨어 개발 경로 선정 알고리즘은 생명주기모형, 소프트웨어 개발 방법론, 프로젝트 규모와 난이도를 고려하며 선정한다. 선정된 항목에 의하여 소프트웨어 개발 작업과 산출물을 추출한다. 그러면 프로젝트 번호, 명칭과 함께 새로운 자료저장소가 형성된다.

3.6 소프트웨어 개발 선정 프로세스 조정

자료 저장소에 저장된 소프트웨어 개발 작업과 산출물들을 전체조회, 단계별로 조회, 영역별로 조회, 단계 · 영역별로 조회하여 작업과 산출물을 수정, 삭제, 삽입할 수 있다. 프로



(그림 3) 소프트웨어 개발 경로 프로세스 조정 절차 흐름도

<표 10> 프로젝트 난이도와 규모 선정에 따른 소프트웨어 개발 프로세스 선정 알고리즘

```
Select_Output_R_as_Project_Area_Complexity_Size(d_area, d_lsize, d_msize, s_ssize,
p_complexity, p_size, Output_R)
DO CASE
    Decide_Project_Size_as_Complexity(p_complexity, p_size, p_con_size)
    CASE d_area = "Strategy"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Strategy" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Strategy" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Strategy" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "Business"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Business" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Business" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Business" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "Process"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Process" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Process" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Process" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "Object"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Object" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Object" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Object" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "Component"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Component" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Component" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Component" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "Data"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Data" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Data" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Data" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "Interface"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "Interface" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "Interface" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "Interface" and d_ssize = "Y" or "O"
    ENDCASE
    CASE d_area = "IT"
    DO CASE
        CASE p_con_size="Large"
            Select Output_R where d_area = "IT" and d_lsize = "Y" or "O"
        CASE p_con_size="Middle"
            Select Output_R where d_area = "IT" and d_msize = "Y" or "O"
        CASE p_con_size="Small"
            Select Output_R where d_area = "IT" and d_ssize = "Y" or "O"
    ENDCASE
    ENDCASE
ENDCASE
```

젝트 경로 프로세스 조정의 경우 <표 1>의 아. 기술위협

자. 조직 정책, 차. 소프트웨어 유지보수 전략에 의하여 소프트웨어 개발 작업과 산출물 조정이 필요하다면 조정할 수 있다. (그림 3)은 소프트웨어 개발 경로 프로세스 조정 절차 흐름도이다[16].

4. 적용 및 평가

4.1 적용사례

본 논문에서는 제시한 소프트웨어 개발 경로를 커스터마이징하는 방법을 반영한 개발 경로 선정 시스템(Tool@Roadmap)을 웹 기반 시스템 개발 프로젝트에 적용한다.

웹 기반 시스템 개발 프로젝트는 서울시 정보 포털 사이트 구축 시스템 적용사례다. Web@RoadMap 방법론 적용 경로 사례인 웹 기반 시스템 개발 프로젝트 단계 활동별 산출물[17, 18]과 개발 경로 선정 시스템(Tool@RoadMap)을 적용한 단계 활동별 산출물과 비교한 결과는 <표 11>의 차이점과 같다.

<표 11>은 웹 기반 시스템, 단계적 접근법, 중형 시스템을 선택한 후 분석, 설계, 구축 단계의 필수 출력자료와 선택사항은 시스템 아키텍처 설계 활동 출력자료를 추출한 것이다. 차이점 중 X 표시한 것은 적용사례에 없는 것으로 다른 산출물과 합쳐진 것으로 판명된다. ① 양식명세서와 ② 현행업무절차서는 업무 기술서에 포함이 되며, ③ 웹 프로세스 정의서는 설계 단계에 웹 프로그램 정의서를 작성하는 것으로 대체하고, ④ 프로그램 일람은 응용 시스템 구조도에 합하며, ⑤ 항해도 ⑥ 멀티미디어 설계서 ⑦ 스토리보드(storyboard)는 사이트 맵과 웹 사이트 페이지 설계서로 대체한다. ⑧ 코드 설계서는 테이블 정의서 또는 프로젝트 관리 표준에 포함이 되었으며, ⑨⑩⑪⑫ 물리설계 활동의 출력자료는 논리설계 활동의 동일명칭의 출력자료로 대체하고, 물리설계의 특수성이 있다면 논리설계 출력자료를 보강한다. ⑬ 공통 모듈 명세서는 웹 프로그램 정의서로 대체되었다. 다른 변경이 있다면 명칭을 프로젝트 특성에 맞게 수정한 것이다. 이들의 작업이 합당하다고 판명하여 경로 프로세스 조정 작업에서 조정을 한 것이다.

4.2 평가

소프트웨어 프로세스 평가를 위한 국제 표준으로는 ISO/IEC 15504[5]가 대표적인 표준 중에 하나이다. 1991년 6월

<표 11> 시스템에 의한 웹 기반 시스템 개발 프로젝트 단계 활동별 산출물 내역서

단계	활동	출력자료	작업	차이점	
요구분석	사용자 요구 사항 정의	개발 계획서	개발계획서 작성		
		양식명세서	사용자요구사항 정리	X ①	
		요구사항기술서	사용자요구사항 정리		
		현행 시스템 사양서	현행 시스템 분석	명칭 수정	
	엔터티 모형 구성	현행 업무절차서	사용자요구사항 분석	X ②	
		엔터티 정의서	엔터티 모형 정의	명칭 수정	
		컨텐츠 정의서	컨텐츠 및 서비스정의		
	프로세스 모형 구성	엔터티 연관도	엔터티 상세 분석		
		프로세스 계층도	프로세스 모형 정의		
		프로세스 흐름도	프로세스 모형 정의		
엔/프 연관분석	웹 프로세스 정의서	프로세스 모형 정의	X ③		
	프로세스/엔터티 매트릭스	프로세스/엔터티 관계 분석			
사용자 인터페이스	디자인 컨셉 정의서	사용자 인터페이스 분석			
	웹 시나리오 정의서	웹 시나리오 작성			
설계	논리설계 (데이터 설계, 웹 어플리케이션 설계)	테이블 일람(논리)	논리 데이터 설계	웹 프로그래밍 정의서 누락	
		테이블 구조도(논리)	논리 데이터 설계		
		테이블 정의서(논리)	논리 데이터 설계		
		프로그램 일람(기본,조회)	프로그램 전환설계		X ④
		응용 시스템 구조도	프로그램 전환설계		
	인터페이스 설계	사이트 맵	웹 사이트 구조 및 향해 설계		
		향해도	웹 사이트 구조 및 향해 설계	X ⑤	
		멀티미디어 설계서	웹 페이지 및 멀티미디어 설계	X ⑥	
		Storyboard	웹 페이지 및 멀티미디어 설계	X ⑦	
		웹 사이트 페이지 설계서	웹 페이지 및 멀티미디어 설계		
	물리 설계	코드설계서	DB/File 및 코드 설계	X ⑧	
		테이블 일람(물리)	성능개선 구조 조정	X ⑨	
		테이블 정의서(물리)	성능개선 구조 조정	X ⑩	
		테이블 구조도(물리)	성능개선 구조 조정	X ⑪	
		데이터베이스 명세서	성능개선 구조 조정	X ⑫	
시스템 아키텍처 설계	공통모듈 명세서	프로그램 구조 및 웹 모듈 설계	X ⑬		
	사용자 권한 테이블	웹 보안기능 설계	선택 사항		
	웹 서버 계획서	시스템 구성 설계			
	인트라넷 네트워크 계획서	시스템 구성 설계			
인트라넷 클라이언트 계획서	시스템 구성 설계				
구축	구현	원시코드(웹 포함)	프로그램 구현		
	지침서개발	사용자 지침서 운영자 지침서	사용자 지침서 완성 운영자 지침서 완성		

제4차 ISO의 SC7 총회의 연구승인 시작으로부터 출발된 ISO/IEC 15504는 소프트웨어 개발 프로세스에 대한 계획, 관리, 감시, 업그레이드를 위한 능력평가와 개선에 있다. ISO/IEC 15504(일명 SPICE)는 SEI(Software Engineering Institute)의 CMMI(Capability Maturity Model Integration), Esprit의 Bootstrap 등의 평가 방법들과는 달리 소형에서부터 대형 소프트웨어 개발 공정까지 적용할 수 있는 특징이 있다.

소프트웨어 개발 경로 선정 시스템 평가를 위해 <표 1>의 소프트웨어 개발 프로세스 선정에 사용되는 영역 특성의

<표 12> 소프트웨어 프로세스 평가

프로세스 관리 항목	PA 1.1 수행속성	PA 2.1 수행리속성	PA 2.2 작업산출물관리	PA 3.1 프로세스 정의	PA 3.2 프로세스 전개	PA 4.1 프로세스 측정	PA 4.2 프로세스 통제	PA 5.1 프로세스 혁신	PA 5.2 프로세스 최적화	종합평정
ENG1.1 시스템요구분석 및 설계	F	F	F	F	L	P	P	N	N	3
ENG1.2 S/W요구분석 및 설계	F	F	F	F	F	F	F	N	N	4
ENG1.3 S/W설계	F	F	F	F	F	F	F	N	N	4
ENG1.4 S/W구축	F	F	F	F	L	L	P	N	N	3
ENG1.5 S/W통합	F	F	F	F	L	L	P	N	N	3
ENG1.6 S/W시험	F	F	F	F	L	L	P	N	N	3

각각의 프로세스에 대해 관리 활동 항목별 평가 결과는 다음 <표 12>와 같이 나타나 있다.

그 결과를 고찰해 보면 시스템 요구분석 및 설계 항목에서는 사용자 요구 명세서를 입력 산출물로 하여 수행속성, 수행관리속성, 작업산출물 관리, 프로세스 정의를 완전히 달성한 반면에, 프로세스 전개는 최소한의 요구사항을 달성하며, 프로세스 측정 프로세스 통제에서는 일부 보완이 필요하며, 프로세스 혁신, 최적화 개선 항목에서는 정의된 속성을 달성한 결과가 발견되지 않아서 추후 소프트웨어 개선작업 시 보완이 필요하다. 소프트웨어 개발 경로 선정 시스템 사용 전 과 비교하면 레벨 판정이 레벨 2에서 레벨 3으로 개선됨을 알 수 있다. 즉 소프트웨어 프로세스가 확립된 수준으로, 잘 정의된 프로세스를 사용하여 프로세스를 수행하고 관리하는 상태로 평가되었다

4.3 효과

기존의 소프트웨어 개발 경로 선정 방법과 비교하여 볼 때, 개선된 소프트웨어 개발 경로 선정 알고리즘은 특징별로 아래와 같은 효과를 가져온다.

4.3.1 소프트웨어 개발 프로젝트 경로 프로세스의 용이한 선정으로 공수 절감

기존의 소프트웨어 개발 경로 프로세스 선정 방법은 방법론 전문가의 과거 경험과 지식에 의존하여 경로 프로세스를 선정하는 것을, 개선 소프트웨어 개발 경로 프로세스 선정 방법은 프로젝트 관리자나 시스템 분석가에 의하여 프로젝트 환경과 특성 영역의 선정 항목을 선택하여 용이하게 경로 프로세스를 선정하고 경로 프로세스의 필수 선택항목에 의하여 용이하게 조정할 수 있다.

또한 기존에는 방법론 전문가가 선정된 작업과 산출물의 표준을 만들고 프로젝트 관련 당사자간의 합의를 거쳐야 하며 이를 위하여 중 규모 프로젝트의 경우 프로젝트 기간동안 1명의 전담요원이 필요하며, 대규모 프로젝트의 경우에는 많은 공수가 필요하다. 그러나 개선된 알고리즘을 반영한 개발 경로 선정 시스템에 의하여 산출물 작성방법과 기법을 제공함으로써 용이하게 프로젝트 관련자들의 합의점을

도출하고 개발자 스스로 작업을 할 수 있으므로 프로젝트 일정 단축과 방법론 전문가의 활용 공수를 줄일 수 있다.

4.3.2 프로젝트 환경과 특성에 적합한 경로 프로세스 선정 개발비용 절감

기존 소프트웨어 개발 경로 선정은 동일한 프로젝트라 하더라도 방법론 전문가에 따라 경로 프로세스 편차가 있을 수 있으며, 소형 중형 대형 프로젝트에 적합하게 프로세스를 선정하는 것이 어려워 프로젝트에 적합한 품질과 비용의 확신이 없으나, 개선 소프트웨어 개발 경로 프로세스 선정 모형은 개발 경로 선정 시스템에 의하여 프로젝트 환경과 특성에 맞는 경로 프로세스를 선정함으로써 동일한 프로젝트의 경우는 동일한 경로 프로세스가 선정된다. 그리고 프로젝트 환경과 특성 선정 항목에 의하여 프로젝트에 적합한 경로 프로세스를 선정하게 하여 프로젝트 환경과 특성에 맞는 품질과 적합한 개발비용이 소요되게 한다.

4.3.3 ISO/IEC 15504의 우수한 레벨 판정으로 품질평가 향상

본 논문에서는 제시한 소프트웨어 개발 경로를 커스터마이징하는 방법을 반영한 개발 경로 선정 시스템을 사용하면 기존에는 개발경로 선정이 경험적인 것에 비하여 잘 정의된 프로세스를 사용하여 프로세스를 수행하고 관리하는 상태로 판정되어 레벨 3의 수준으로 평가되고 그리고 계속적인 개선작업을 실시한다면 좀더 상위 레벨 수준으로 향상시킬 수 있음을 기대 할 수 있다.

5. 결론 및 연구방향

본 논문은 소프트웨어 개발 프로젝트 환경과 특성에 맞는 경로 프로세스 선정을 용이하게 하는 것이다.

개발 경로 선정 기준은 ISO/IEC TR 15271 Information Technology-Guide for ISO/IEC 12207(Software Life Cycle Processes), ISO/IEC 15504 Information technology-Process assessment의 프로세스 내용에 기반한 10개의 영역 특성이다. 10개의 영역 특성에 대하여 항목을 정의한다. 영역 특성과 항목은 프로젝트 환경과 특성을 반영한 것이며, 프로젝트 개발 경로 선정 항목이다.

개발 경로 선정 항목을 선택하여 소프트웨어 프로젝트 최적 개발 경로를 선정하는 알고리즘을 제시한다. 알고리즘은 소프트웨어 개발 생명 주기 선정, 소프트웨어 개발 방법론 선정, 프로젝트 규모 선정, 프로젝트 난이도 선정 항목으로 작성된다.

그리고 선정된 경로 프로세스는 프로젝트 위험, 조직 정책, 유지보수 전략 선정 항목에 의하여 조정되어야 하며, 이를 용이하게 하기 위하여 소프트웨어 개발 프로세스 개선 절차를 이용한다[16].

ISO/IEC 15504에 의한 시스템 평가 방법에 의하여 본 논

문의 소프트웨어 최적 개발 경로 선정 모형을 적용한 개발 경로 선정 시스템이 적정한지 평가 한다. 평가결과는 레벨 3 수준이다. 레벨 3 수준은 소프트웨어 프로세스 확립수준으로 잘 정의된 프로세스를 사용하여 프로세스를 수행하고 관리하는 정도다.

향후 연구 방향으로 개발 경로 선정 시스템 적용을 거듭하여, 개발 경로 선정 시스템 프로세스를 개선한다면 좀더 상위 레벨 수준으로 향상시킬 수 있음을 기대 할 수 있다. 이를 위하여 소프트웨어 개발 경로 프로세스를 체계적으로 표준화하고 관리 적용하여야 한다.

본 논문에서 제안한 개발 경로 선정 시스템을 개선하여 소프트웨어 개발 경로 프로세스가 결정되면, 개발 경로 선정 시스템은 경로 프로세스를 이용하여 프로젝트를 진행할 수 있도록 한다. 개발 경로 선정 시스템은 개발자에게 프로젝트 진행하는데 도움이 되는 정보를 제공한다. 그리고 개발자가 개발 경로 선정 시스템에서 제공되는 정보를 통하여 프로젝트를 용이하게 수행할 수 있고, 프로젝트 관리자에게는 개발자들의 작업 진행결과를 자연스럽게 파악할 수 있는 시스템으로 개선 할 수 있다.

CASE도구 개발의 연구를 지속하여 소프트웨어 개발 및 유지보수가 자동화되도록 하여야 한다.

참 고 문 헌

- [1] 안유환, 안형준, 박성주, "소프트웨어 프로젝트 프로세스의 계획을 지원하기 위한 지식기반 프로세스 라이브러리 접근법", 한국프로젝트 관리 기술회 프로젝트관리기술 논문집, pp.55-72, 2001. 4
- [2] Bomarius, F., Althoff, K.D., and Muller, W. Knowledge Management for Learning Software Organizations. Software Process-Improvement and Practice, 4(2), pp.89-93. 1998.
- [3] Kershberg, L., Gomma, H., Mohan, R. G., and Farrukh, G.A. "PROGEN: A Knowledge-based System for Process Model Generation, Tailoring and Reuse", ISSE-TR96-05, George Mason University. 1996.
- [4] ISO/IEC TR 15271 Information Technology-Guide for ISO/IEC 12207(Software Life Cycle Processes)
- [5] ISO/IEC 15504-1 Information technology - Process assessment-Part 1: Concepts and vocabulary
- [6] Lam, W., and McDemid, J.A, "A Summary of Domain Analysis Experience By Way of Heuristics", Proceedings of SSR'97(pp.54-64), MA, USA, 1997
- [7] Baumert, J., "Experiences Developing and Deploying a Corporate-wide Process Asset Library", Proceedings of 1996 SEPG conference(pp.20-23), Atlantic City, NJ, 1996
- [8] Zahran, S. Software process Improvement: Practical Guidelines for business success. Addison-Wesley, 1998
- [9] Heninger, S., Lappala, K., and Raghavendran, A., "An Organizational Learning Approach to Domain Analysis", Proceedings of 17th international conference on Software Engineering, (pp.95-104), Seattle, Washington USA. 1995
- [10] 정기원, 조병호, "소프트웨어 생명주기개념 선정방법과 생명

주기모델 설계절차”, 정보과학회 논문지(B), 제22권, 제9호, pp.1321-1332 1995년 9월

- [11] 정병권, 송재형, “폭포수 모형 소프트웨어 개발 프로세스와 프로젝트 관리 프로세스를 이용한 소프트웨어 개발 생명주기 모형 적용”, 한국프로젝트 관리 기술회 프로젝트관리기술 논문집, pp.23-32, 2002. 8.
- [12] Reinhard Budde, Kalheinz Kautz, Karin Kuhlenkamp, Heinz Züllighoven, “Prototyping An Approach to Evolutionary System Development”, Springer-Verlag, pp.34-39, 1991.
- [13] 주복규, 김영철, “소프트웨어 제품 군을 개발하기 위한 점진적 방법”, 정보처리학회논문지D, 제10-D권 제4호, pp.697-708, 2003. 8.
- [14] Reinhard Budde, Kalheinz Kautz, Karin Kuhlenkamp, Heinz Züllighoven, “Prototyping An Approach to Evolutionary System Development”, Springer-Verlag, pp.67-71, 1991
- [15] James Martin, “Rapid Application Development”, MACMILLAN PUBLISHING COMPANY, 1991.
- [16] 정병권, 윤석민, “구조적 또는 정보공학 소프트웨어 개발 방법론 개선”, 정보처리학회논문지D, 제9-D권 제6호, pp.1083-1090, 2002, 12.
- [17] 정병권, 김동수, 송재형, 황종선, “웹기반의 시스템 개발 방법론 개발과 적용-분석 및 설계 단계를 중심으로”, 정보과학회 논문지: 컴퓨팅의 실제분야, pp.156-166, 2002. 4.
- [18] Byung-Kwon Jung, Dong-Soo Kim, Seok-Min Yoon, Gyu-Sang Shin, Chong-Sun Hwang, “Development and application of a model for analysis and design phases of Web-based system development” Journal of Science in China Series F, Volume 46(2003)



정 병 권

e-mail : bkjung@doowon.ac.kr

1977년 고려대학교 수학과(학사)

1986년 연세대학교 산업대학원 전자계산학과 (공학석사)

1990년 정보처리기술사 취득

1999년 정보시스템 감리인 취득

2002년 고려대학교 대학원 전산학과(이학박사)

1979년~1998년 LG 전선(주), (주)LG-EDS시스템 기술연구부문 전문부장

1998년~현재 두원공과대학 테크노경영계열 교수

관심분야 : 소프트웨어 공학(소프트웨어 개발 방법론), 경영정보, 프로젝트 관리



윤 석 민

e-mail : smyoon@dongyang.ac.kr

1981년 연세대학교 전자공학과(학사)

1986년 연세대학교 대학원 전자공학과 (공학석사)

2002년 충북대학교 대학원 컴퓨터공학과 (공학박사)

1986년~현재 동양공업전문대학 소프트웨어정보과 교수

관심분야 : 소프트웨어공학, 멀티미디어 시스템