

## 프로젝트 난이도와 규모에 따른 소프트웨어 개발 프로세스

윤석민\*, 김인식\*\*

### Software Development Process for Project Complexity and Size

Seok-Min Yoon \*, In-Sik Kim \*\*

#### 요 약

본 논문의 목적은 디지털 저작권에 기반기술로 이용될 수 있는 워터마킹 알고리즘 구성에 대한 소프트웨어 프로세스 심사에 있다. 소프트웨어 프로젝트에 수반되는 작업 프로세스들이나 관련된 작업 자원들은 서로 다른 복잡성을 갖는다. 경험 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 개발 방법론에 거론되는 전체 항목을 수행하려다 보면 많은 개발비용이 소요되어 방법론 적용에 문제가 될 수 있다. 이러한 문제점들을 해결하기 위한 방법으로 소프트웨어 프로세스와 관련 지식들을 재활용하는 접근 방법들이 제시 되어왔다. 본 논문에서는 프로젝트의 환경과 특성 요인들을 분석하여 그들에 맞는 프로세스들을 선정하는 방법을 제시한다. 프로젝트 환경과 특성에 맞는 개발 경로 선정을 용이하게 하기 위하여 ISO/IEC 15271, ISO/IEC 12207 및 ISO/IEC 15504의 프로세스 내용에 준하여 영역 특성을 분류하여 프로젝트 개발 경로 선정 항목을 정하여 이를 본 논문에서의 적용범위로 한다.

#### Abstract

In this thesis, for the assessment of software development process for project complexity and project size : one of digital watermarking algorithm is selected and examined. Digital watermarking provides a solution to illegal copying of digital contents and has many other useful applications, including web broadcast monitoring, transaction tracking, authentication, copy control and device control. This thesis focused on the method for customizing software development path, considering the project environments and characteristics. The selection standard of software development path is composed of process items, based on the ISO/IEC 15271 Information Technology Guide for ISO/IEC 12207 (Software Life Cycle Process) and ISO/IEC 15504 Information technology - Process assessment. To evaluate the algorithm, a system for selecting development path, which reflected algorithm, was examined.

▶ Keyword : Software process, Digital watermarking, Project environment, Process assessment

• 제1저자 : 윤석민 교신저자 : 김인식

• 투고일 : 2009. 06. 11, 심사일 : 2009. 06. 23, 게재확정일 : 2009. 06. 23.

\* 동양공업전문대학 소프트웨어정보과 교수 \*\* 경북전문대학 컴퓨터정보과 교수

※ 본 연구는 2007년도 동양공업전문대학 학술연구과제 지원에 의하여 이루어졌습니다.

## I. 서론

본 논문의 목적은 디지털 저작권에 기반기술로 이용될 수 있는 워터마킹 알고리즘 구성에 대한 소프트웨어 프로세스 심사에 있다. 소프트웨어 프로젝트에 수반되는 작업 프로세스들이나 관련된 작업 자원들은 서로 다른 복잡성을 지니며, 이를 관리하는 것은 개발 담당자의 개인들의 지식과 경험에 의존하여 왔다. 따라서 이러한 형태의 관리 과정은 매우 많은 시간이 소모되며 잦은 오류가 발생된다. 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 개발 방법론에 거론되는 전체 항목을 수행하려다 보면 많은 개발비용이 소모되어 방법론 적용에 문제가 될 수 있다. 이러한 문제점들을 해결하기 위한 방법으로 소프트웨어 프로세스와 관련 지식들을 재활용하는 접근 방법들이 제시 되어왔다. 이러한 방법들은 주로 재활용 정보저장소의 내용과 구조, 저장 및 검색 방법들에 연구의 초점을 맞추어 왔다. 이러한 접근 방법은 새로운 프로젝트를 위한 소프트웨어 프로세스를 조정해 내는데 특징이 있다.

## II. 관련 연구

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 특성에 맞는 개발 경로 선정을 용이하게 하기 위하여 ISO/IEC 15271, ISO/IEC 15504의 프로세스 내용에 준하여 영역 특성을 분류하여 프로젝트 개발 경로 선정 항목을 정하여 이를 본 논문에서의 적용범위로 한다. 소프트웨어 개발 경로를 커스터마이징하는 방법을 적용하고 평가하기 위하여 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 시스

템을 국제표준 ISO/IEC 15504의 평가방법을 이용하여 평가를 진행한다[1].

소프트웨어 개발 프로젝트 경로 프로세스를 선정하는 방법으로 프로젝트의 과거자료를 이용함으로 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 포함한다. 그러나 이러한 연구는 도메인의 특성으로부터 소프트웨어 프로세스를 추출할 수 있도록 해주었으나 다양한 응용 도메인에 적용하기에는 한계점을 가지고 있다. 프로젝트의 과거 경험을 재활용하기 위한 정보 저장소에 관한 연구로는 PAL(Process Asset Library)과 같은 대표적인 연구들도 있으며 그 프로토타입도 제시되었다. 또한 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 적응시키는 규격을 제시하지 못하고 있다.

적용되는 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 15721 Information Technology Guide for ISO/IEC 12207(Software Life Cycle Processes), ISO/IEC 15504 Information Technology Software Process Assessment의 프로세스 내용을 기반으로 한 영역 특성의 선정 항목들로 프로젝트 규모, 난이도에 적합한 소프트웨어 개발 프로세스를 적용한다[2][3].

### III. 본 론

#### 1. 평가대상 프로젝트 선정

소프트웨어 개발 프로젝트의 규모에 따라 적합하게 적용하지 않고, 소규모의 프로젝트도 소프트웨어 개발 방법론에 거론되는 전체 항목을 수행하려다 보면 많은 개발비용이 소요되어 방법론 적용에 문제가 될 수 있다. 따라서 본 논문에서 평가대상은 비선형 동영상 편집기에 워터마킹을 적용하는 프로젝트로 한다. 워터마킹(watermarking)은 정보 은닉기법의 한 종류로서 원시 영상에 저작권 정보인 워터마크와 비밀키를 함께 저장시킨다. 그러나 삽입된 워터마크 정보는 추가적인 영상 편집을 통해 장면 전환 효과 등의 삽입과 여러 방법을 이용하여 변조시킨 영상에서도 견고하게 검출될 수 있어야 한다. 효율적인 워터마킹을 위해서는 화질의 열화 방지와 은닉 정보의 견고성 유지가 필수적인 조건이다. 현재 동영상을 위한 워터마킹 기법으로는 주파수 확산(spread spectrum)워터마킹과 움직임 벡터(motion vector) 워터마킹 기법을 들 수 있다. 그러나 주파수 확산 워터마킹은 편집해야 할 동영상 시퀀스의 대부분을 차지하는 P, B 프레임에는 적용하기가 어렵고, 움직임 벡터를 이용한 워터마킹은 I 프레임에는 정보를 저장 할 수 없다는 특징이 있다[4][5].

#### 2. 워터마킹 알고리즘

본 연구에서 제시한 다중 워터마킹 알고리즘은 동영상 스트림의 I 프레임에 대해서는 주파수 확산 워터마킹 기법을 적용하고, P, B 프레임에 대해서는 움직임 벡터 워터마킹 기법을 적용한다. 즉, 주파수 확산 워터마킹 기법의 장점인 에러에 대한 강인성과 움직임 벡터 워터마킹 기법의 장점인 실시간성을 결합시킨 알고리즘을 사용한다.

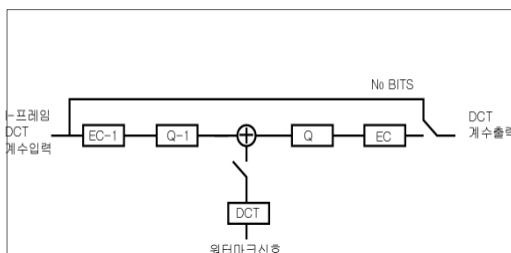


그림 1. I 프레임의 워터마킹 과정  
Fig 1. Watermarking of I Frame

〈그림 1〉과 같은 워터마킹 과정에서 의사 잡음(pseudo noise) 시퀀스는 사용자 암호(user key)에 의해서 생성된다. 생성된 의사 잡음 시퀀스를 이용하여 삽입하고자 하는 저작권 정보를 확산시킨다. 편집된 동영상 시퀀스에 직접 워터마크를 삽입하기 위해서는 우선 DCT를 수행한다. 그리고 I 프레임의 계수 입력과 합하여 DCT 계수를 출력한다. 이 결과 값을 다시 양자화하여 가변길이 부호화(variable length coding)를 통하여 워터마크 삽입을 한다. P 프레임 또는 B 프레임의 경우에는 〈그림 2〉와 같은 과정을 통해 동영상 시퀀스를 디코딩 한다. P, B 프레임의 워터마킹 과정은 P 프레임과 B 프레임을 입력으로 하여 프레임 데이터가 아닌 움직임 벡터를 구한 후에 저작권 정보가 있는 워터마크를 삽입한다.

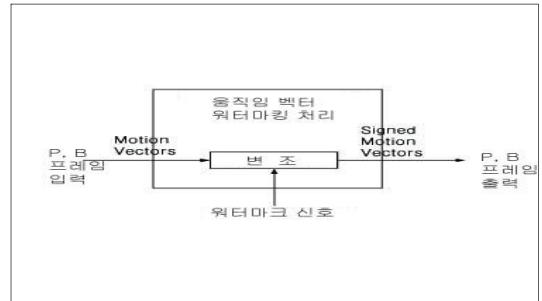


그림 2. P, B 프레임의 워터마킹 과정  
Fig 2. Watermarking process of P, B frame

이상과 같이 하여 주파수 확산 워터마킹 기법의 장점인 에러에 대한 강인성과 움직임 벡터 워터마킹 기법의 장점인 실시간성을 결합시킨 알고리즘을 사용한다. 즉 두 가지 워터마킹 방법의 장점만을 적용하여 웹 캐스팅, 저작권 파괴 공격 및 영상 손실 등의 각종 에러 환경에서도 강인한 특성을 갖는 워터마킹 기법이 된다.

#### 3. 프로젝트 난이도 선정 프로세스

프로젝트 난이도 선정 프로세스는 〈표1〉 프로젝트 난이도에 따른 프로세스 선정 알고리즘을 기본으로 적용한다. 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[6][7].

표 1. 프로젝트 난이도에 따른 프로세스 선정 알고리즘  
Fig 1. Process selection algorithm for project complexity

```
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 P_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
```

5	업무 배경도 (context diagram)	갑/을	1	1	1
6	기능 분해도	을/갑	1	1	1
7	주요 엔터티 연관도	을/갑		1	1
8	웹 주요 엔터티 목록 (주요 엔터티 리스트)	을/갑		1	1
9	원인 및 동기 리스트	갑		1	1
10	웹 사이트 개요서 (응용 시스템 개요서)	을/갑		1	1
11	데이터 개요서	을/갑		1	1
12	정보기술 개요서 (정보시스템 개요서)	을/갑		1	1
13	시스템 사양서	을/갑		1	1
14	정보관리 개요서	을/갑		1	1
15	프로젝트 산출물 내역서	을/갑	1	1	1
16	개발 계획서 (프로젝트 계획서)	을/갑	1	1	1
17	보고 계획서	을/갑		1	1
18	단계 점검 결과서	을/갑		1	1

4. 프로젝트 규모 선정 프로세스

프로젝트 규모 선정에 따른 소프트웨어 개발 프로세스 선정방안은 작성된 자료사전에서 보면 프로젝트 규모 항목이 소규모, 중규모, 대규모로 구분되어 작업과 산출물을 필수 선택으로 선정할 수 있도록 한다. 이를 지원하기 위하여 정보처리 프로젝트 규모별로 산출물을 선정할 도표는 <표 2>와 같다 [8][9].

표 2. 프로젝트 규모에 따른 산출물 선정  
Fig 2. Output document for Project size

단계	단계 별 No	문서명	작성 주관	1~2	3~6개월	9개월
				월소 규모 선택	이내 중규모 필수	이상 대규모 선택
개발 계획	1	프로젝트 개요서	갑/을		1	1
	2	비즈니스 모델 정의서	갑	1	1	1
	3	웹 사업요구 정의서	갑	1	1	1
	4	웹 요구기능 목록	갑	1	1	1

<표 2>에 의해서 소프트웨어 개발 방법론 자료저장소에 프로젝트 규모표시를 할 수 있다. 이는 소프트웨어 난이도와 연계하여 <표 3> 프로젝트에 따른 규모 선정 알고리즘에 반영한다.

표 3. 프로젝트 난이도와 규모 선정에 따른 소프트웨어 개발 프로세스 선정 알고리즘  
Fig 3. Software development process selection algorithm for Project Complexity and Size

```
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 = "Small"
Select Output_R where d_area = "Strategy" and
d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "Business" and
d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "Process" and
d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "Object" and
d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "Component"
and d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "Data" and
d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "Interface" and
d_ssize = "Y" or "O"
END CASE
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 = "Small"
  Select Output_R where d_area = "IT" and
d_ssize = "Y" or "O"
END CASE
END CASE

```

프로젝트 경로 프로세스 선정 처리는 앞에서 주어진<표 3>의 알고리즘을 동시에 만족하는 작업과 산출물을 선정하는 것이다. 소프트웨어 개발 경로 선정 알고리즘은 생명주기 모형, 소프트웨어 개발 방법론, 프로젝트 규모와 난이도를 고려하며 선정한다. 선정된 항목에 의하여 소프트웨어 개발 작업과 산출물을 추출한다. 그러면 프로젝트 번호, 명칭과 함께 새로운 자료저장소가 형성된다. 자료 저장소에 저장된 소프트웨어 개발 작업과 산출물들을 전체조회, 단계별로 조회, 영역별로 조회, 단계영역별로 조회하여 작업과 산출물을 수정, 삭제, 삽입할 수 있다. 프로젝트 경로 프로세스 조정의 경우 기술위험, 조직 정책, 소프트웨어 유지보수 전략에 의하여 소프트웨어 개발 작업과 산출물 조정이 필요하다면 조정할 수 있다. <그림 3>은 소프트웨어 개발 경로 프로세스 조정 흐름도이다. 개발경로 프로세스 조회를 시작으로 정해진 선정 알고리즘을 통해 반복적인 검토가 이루어지는 과정이 나타나있다.[10][11].

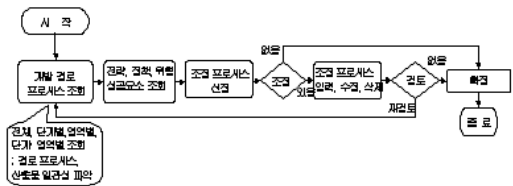


그림 3. 소프트웨어 개발 경로 프로세스 조정 흐름도  
Fig 3. Software development path and process chart

### IV. 적용 및 평가

한편 소프트웨어 개발 경로를 커스터마이징하는 방법을 반영하고 시스템 개발 프로젝트 단계 활동별 산출물과 개발 경로 선정 시스템을 적용한 단계 활동별 산출물과 비교한 결과는 <표 4>과 같다

표 4. 프로젝트 단계별 산출물  
Table 4. Output of project stage

단계	활동	출력자료	작업	비고
계획	계획	개발 계획서	개발계획서 작성	
	사용자 요구사항 정의	양식명세서 요구사항기술서 현행 시스템 사양서 현행 업무절차서	사용자요구사항 결정 사용자요구사항 정리 현행 시스템 분석 사용자요구사항 분석	
요구 분석	엔터티 모형구성	엔터티 정의서 컨텐츠 정의서 엔터티 연관도	엔터티 모형 정의 컨텐츠 및 서비스정의 엔터티 상세 분석	
	프로세스 모형구성	프로세스 계층도	프로세스 모형 선택	
		프로세스 흐름도	프로세스 모형 정의	

	엔/프 연관분석	웹 프로세스 정의서 프로세스/엔터티 매트릭스	프로세스 모형 결정 프로세스/엔터티 관계 분석	
	사용자 인터페이 스	디자인 컨셉 정의서 웹 시나리오 정의서	사용자 인터페이스 분석 웹 시나리오 작성	
설계	논리설계 (데이터 설계, 웹 어플리케이션설계 )	테이블 일람(논리)	논리 데이터 조사	웹프 로그 램정 의서
		테이블 구조도(논리)	논리 데이터 설계	
		테이블 정의서(논리)	논리 데이터 설계	
		프로그램 일람(기본,조회)	프로그램 전환설계	
		응용 시스템 구조도	프로그램 전환설계	
	인터페이 스 설계	사이트 맵	웹 사이트 구조 및 향해 조사	
		향해도	웹 사이트 구조 및 향해 설계	
		멀티미디어 설계서	웹 페이지 및 멀티미디어 설계	
		Storyboard	웹 페이지 및 멀티미디어 설계	
		웹 사이트 페이지 설계서	웹 페이지 및 멀티미디어 설계	
	물리설계	코드설계서	DB/File 및 코드 설계	
		테이블 일람(물리)	성능개선 구조 조사	
테이블 정의서(물리)		성능개선 구조 조정		
테이블 구조도(물리)		성능개선 구조 조정		
데이터베이스 명세서		성능개선 구조 조정		
공통모듈 명세서		프로그램 구조 및 웹 모듈 설계		
시스템 이카택처 설계	사용자 권한 테이블	웹 보안기능 설계	선택 사항	
	웹 서버 계획서	시스템 구성 조사		
	인트라넷 네트워크 계획서	시스템 구성 설계		
	인트라넷 클라이언트 계획서	시스템 구성 설계		
구현	구현 원시코드(웹 포함)	프로그램 구현		
	지침서 개발	사용자 지침서 운영자 지침서	사용자 지침서 완성 운영자 지침서 완성	

본 논문에서는 제시한 소프트웨어 개발 경로를 커스터마이징하는 방법을 반영한 개발 경로 선정 시스템 평가를 위해 체크리스트를 작성하고, 대표적인 주요 프로세스에 대해 프로세스 관리 활동 항목별 프로세스 평가 결과는 다음 <표 5>(<표 6>)과 같이 나타나 있다. 특히 <표5>에서 최상위 수준에 대한 시스템구조설정을 위한 프로세스는 입력산출물이 요구되지 않지만 출력산출물을 근거로 관련 프로세스활동을 평가 하게 된다. <표6>에서 알 수 있듯이 프로세스 관리 활동 항목 중에서 ENG1.2 S/W 요구분석 및 설계항목이 상대적으로 양호한 평가가 측정되었다.

표 5. ENG 1.1 시스템 설계 프로세스의 체크리스트  
Table 5. ENG 1.1 Check list of system design process

번호	내 용	입력 산출물	출력 산출물	평가 방법
1	소프트웨어 요구사항 식별	사용자 요구 명세서 소프트웨어 필요성평가서	소프트웨어 요구명세서	입력 산출물 및 출력 산출물, 관련 프로세스 활동
2	시스템 요구 사항 분석	유지보수 요구 명세서	인터페이스 명세서	입력 산출물 및 출력 산출물, 관련 프로세스 활동
3	최상위 수준 시스템구조 설정	NA	시스템설계 / 구조도	출력 산출물, 관련 프로세스 활동
4	시스템 구조의 구성요소 할당	NA	NA	관련 프로세스 활동
5	요구사항 우선 순위별 시스템 릴리즈 계획	사용자 집단 분석	릴리즈전략 / 계획	입력 산출물 및 출력 산출물, 관련 프로세스 활동
6	시스템 요구사항 전달체계	변경요청서	NA	입력 산출물 관련 프로세스 활동
7	사용자Needs와 시스템 요구사항 간의 관계 추적	고객 요청기록	추적성기록 / 매핑	입력 산출물 및 출력 산출물, 관련 프로세스 활동

표 6. 프로세스 평가 테이블 구성  
Table 6. Table formation of process evaluation

프로세스 관리활동 항목	PA1.1 수행속성	PA2.1 수행관리속 성	PA2.2 작업산출 물관리	PA3.1 프로세스 정의	PA3.2 프로세스 전개	PA4.1 프로세스 측정	PA4.2 프로세스 통제	PA5.1 프로세스 혁신	PA5.2 프로세스 최적화	종합레벨 판정
ENG1.1 시스템 요구분석 및 설계	F	F	F	F	L	P	P	N	N	3
ENG1.2 S/W 요구분석 및 설계	F	F	F	F	F	F	F	N	P	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	P	3
ENG1.6 S/W시험	F	F	F	F	L	L	P	N	N	3

## V. 결 론

### 1. 성능 시험에 대한 프로세스 평가

원시 영상에 삽입된 워터마크에 대한 정량적인 평가 기준이 필요하다. 즉 저작권 또는 소유권 정보(Ownership information)가 적용된 후, 워터마크의 정보 추출은 동영상을 재생할 때 수행하게 된다. 생성된 영상 스트림을 전부 프레임으로 구성 한 후 워터마크의 존재여부 및 여러 상태와 저작권 공격에 대한 강인성을 평가한다. 적용된 하이브리드(Hybrid) 동영상 워터마킹 알고리즘은 MPEG 스트림의 I-프레임에 대해서는 주파수 확산 워터마킹 기법을 사용하고 P, B-프레임에 대해서는 움직임 벡터 워터마킹을 사용하는 방법으로 주파수 확산 워터마킹의 장점인 에러에 대한 강인성과 움직임 벡터 워터마킹의 장점인 실시간성을 결합한 방법이다. 따라서 삽입된 저작권 정보가 주파수 확산 워터마킹 기법 및 움직임 벡터 워터마킹 기법으로 각각의 I-프레임과 P, B-프레임에 정상적으로 삽입되었고 추출결과를 평가한다. 즉 원영상과 워터마크된 영상의 각프레임 별 PSNR 값과 추출된 워터마크 전보의 비트에러율을 평가해 본다. 또한 동영상의 각 프레임들을 1초 단위로 구분하여 각 시간대 별로 PSNR값을 평균하여 Streaming방식에 의한 웹 캐스팅된 결과에 대한 성능 평가를 실시한다.

### 2. ISO/IEC 15504의 상급 레벨 판정으로 품질 평가 향상

프로세스 레벨은 3개 주요 생명주기 프로세스인 기본(Primary) 프로세스, 지원(Supporting), 조직(Organization)그룹과 5개 범주인 사용자-개발자(Customer-Supplier), 엔지니어링(Engineering), 지원(Supporting), 관리(Management), 조직(Organization) 등 분야의 총40개 총 프로세스로 구성되지만 본 논문의 평가는 소프트웨어의 설계 프로세스인 ENG 1.1에 국한시킨다. 평가 절차는 해당 프로세스의 내용별 입력 및 출력 산출물을 BP(Basic Practices) 및 MP(Management Practice)별로 점검하여 N(Not Achieved), P(Partially Achieved), L(Largely Achieved) 및 F(Fully Achieved)의 능력 레벨기준을 평가한다. 그리고 최종 집계된 소프트웨어 프로세스 등급결정을 Level 0 (불완전 수준), Level 1 (수행수준), Level 2(관리 수준), Level 3 (확립 수준), Level 4 (예측가능 수준) 및 Level 5(최적 수준) 내에서 평가하게 된다. 소프트웨어 개발 경로를 커스터마이징하는 방법을 반영해서 기존에는 개발경로 선정이 경험적인 것에 비하여 잘 정의된 프로세스를 사용하여 프로세스

를 수행하고 관리하는 상태로 판정되어 레벨 3의 수준으로 평가되고 그리고 계속해서 개선적임을 한다면 레벨 4의 수준으로 향상시킬 수 있다.

### 3. 향후 개선 사항

그 결과를 정리해 보면 시스템 요구분석 및 설계 항목에서는 사용자 요구 명세서를 입력 산출물로 하여 수행속성, 수행관리 속성, 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 향후 소프트웨어 개선 작업 시 보완이 필요하다. 종합적 레벨 판정이 레벨 3으로 판정 되었다. 즉 소프트웨어 프로세스가 확립된 수준으로, 잘 정의된 프로세스를 사용하여 프로세스를 수행하고 관리하는 상태로 판정되었다.

## 참고문헌

- [1] ISO/IEC 15271 Information Technology - Guide for ISO/IEC 12207(Software Life Cycle Processes)
- [2] ISO/IEC 15504-1 Information technology - Process assessment Part 1: Concepts and vocabulary
- [3] 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
- [4] 정병권, 윤석민, "구조적 또는 정보공학 소프트웨어 개발 방법론 개선", 정보처리학회 논문지 D 제9-D권 제6호, 1083 - 1090쪽, 2002년 12월
- [5] 김인식, 김종겸, 정순기 "ROLAP 환경에서 집단함수 질의처리를 위한 효율적인 알고리즘" 한국컴퓨터정보학회 논문지 제8권 제3호 , 40 - 46쪽 , 2003년, 9월
- [6] 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
- [7] ISO/IEC TR15271 Information Technology - Guide for ISO/IEC 12207

- [8] ISO/IEC TR15504-1 Information technology - Process assessment
- [9] 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
- [10] Baumert, J., "Experiences Developing and Deploying a Corporate-wide Process Asset Library", Proceedings of 1996 SEPG conference(pp.20-23), Atlantic City, NJ, 1996
- [11] Zahran, S. Software process Improvement : Practical Guidelines for business success. Addison-Wesley, 1998

### 저 자 소 개



윤 석 민

1981년: 연세대학교 전자공학과 학사  
1986년: 연세대학교 전자공학과 석사  
2002년: 충북대학교 컴퓨터공학 박사  
1986년 ~ 현재 : 동양공업전문대학  
소프트웨어정보과 교수



김 인 식

2003년 : 충북대학교 컴퓨터공학과  
공학박사  
1990년 ~ 현재 : 경북전문대학  
컴퓨터정보과 부교수