

도메인 특성에 근거한 웹 시스템 개발 프로세스 : 콘텐츠 중심 도메인 및 데이터 중심 도메인

박 수 용[†] · 임 성 재^{††}

요 약

웹 시스템 개발은 기존의 메인 프레임 또는 클라이언트/서버 환경과는 달리 기술적으로 볼 때, 역동적이고 가변적인 요소들이 강하게 작용하며, 비교적 단기간에 이루어지는 만큼 개발자가 쉽게 적용할 수 있는 실용적이고 효율적인 프로세스는 매우 중요하다. 그러나, 웹 시스템 개발에 대한 기술적인 대안과 구현 언어, 아키텍처 구성에 대한 논의는 적극적인 반면, 표준화된 개발 프로세스와 산출물에 대한 논의는 상대적으로 활발하지 못했다. 본 논문에서는 현행 웹 시스템 개발 프로세스에 대한 문제점들을 분석하고, 이에 대한 대안으로써, 반복과 점증의 개념을 적용한 도메인 특성에 근거한 웹 시스템 개발 프로세스를 제안한다. 도메인의 분류는 웹 시스템 요소와 웹 사이트 유형을 상호연결하여, 총5개의 도메인을 설정하고, 프로젝트에 적용된 2개의 도메인에 대한 워크플로우를 작업내용 및 산출물 중심으로 구체화한다. 또한, 제안하는 프로세스를 인트라넷 시스템 개발에 적용하고, 구성과 요소로 분류된 척도를 통해 기존의 웹 개발 프로세스와 비교 평가한다.

Web System Development Process based on Domain Features : Content centric Domain and Data centric Domain

Soo Yong Park[†] · Sung Jae Yim^{††}

ABSTRACT

As the technical aspect of the web system development is, unlike the existing main frame or client/server environment, greatly influenced by the dynamic and variable factors and carried out in a comparatively short period, practical and efficient process to which the developer may adapt with ease is very much required for the web system development. However, while the technological substitute, language and architecture composition have been much discussed, in connection with web system development growing rapidly, standardized development process and its deliverables have been rarely discussed. This thesis analyzes problems of the existing web system development processes and as the substitutes for them, proposes web system development process based on the domain features derived from the concepts of iteration and increment. For the classification of domains, five domains in total are set by interlocking web system factors and web site types and the workflows for two domains applied to projects are incorporated focused on work content and deliverables. The proposed process applies to the development of intranet system and is compared with the existing web development processes by the measure classified into components and elements.

키워드 : 소프트웨어 개발 프로세스(Software Development Process), 웹 시스템 개발(Web System Development), 워크플로우(Workflow)

1. 서 론

1.1 웹 시스템 개발의 특징 및 프로세스의 필요성

1990년대 중반 이후, 정보시스템의 새로운 패러다임으로 등장한 인터넷 관련 웹 시스템은 기존 정보시스템의 주류를 이루어왔던 메인 프레임 및 클라이언트/서버 환경을 대체하면서 급속도로 발전하고 있다[1]. 웹 시스템은 TCP/IP 및 HTTP 프로토콜, 인터넷에서 접근 가능한 자원의 주소를 일관되게 표현하는 형식을 나타내는 URL(Uniform Resource Locator)

과 HTML이라는 웹 전용의 태그 언어를 표준으로 사용하며, 적용 기술이나 개발 프로세스, 사용자 인터페이스 측면에서도 다른 방식이 요구된다. 다음은 웹 시스템 개발의 주요 특징들이다[2].

- 사용자 중심의 계층화된 네비게이션 구조로 설계된다.
- 화면 디자인의 미려도를 고려해야 한다.
- 웹 페이지 형식의 구현방법이 필요하다.
- 요구사항의 지속적 변경이 불가피하다.
- 객체 지향적 접근방법이 요구된다.
- 비교적 짧은 주기로 개발된다.
- 신속한 유지보수 또는 재개발이 필요할 수 있다.
- 최신 기술의 적용에 따른 위험요인이 증가한다.

※ 본 연구는 한국과학재단 특정기초연구 No.2000-1-303-001-3 지원에 의한 것임.

† 정 회 원 : 서강대학교 컴퓨터학과 교수

†† 정 회 원 : 한진정보통신 경영지원본부 사업지원팀 대리
논문접수 : 2001년 8월 29일, 심사완료 : 2001년 11월 27일

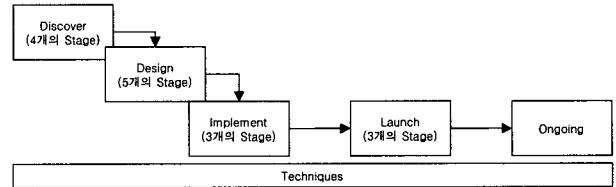
- 웹 프로젝트에 대한 관리 및 개발 표준이 미약하다

현재의 상황을 보면, 웹 아키텍처와 비즈니스 로직을 구현하는 기술 및 프로그래밍 언어, 페이지를 디자인하는 그래픽 요소 등은 지속적인 발전을 거듭하는 반면, 개발 라이프사이클을 제시하고, 경로를 구성하는 프로세스 또는 방법론에 대한 논의는 상대적으로 활발하지 못했다. 이러한 상황으로 인해, 개발 현장에서는 기존에 사용해 온 구조적 기법 및 정보공학 접근방법을 웹 시스템 개발에 적용하고 있고, 웹 프로젝트는 개발자 개인의 경험과 숙련도에 의존한 진행, 임의적인 프로세스 경로(Path) 수정, 개발 프로세스와 산출물의 불일치와 이로 인한 시스템 유지보수 및 확장의 어려움 등을 반복하고 있다[3]. 이러한 웹 개발 문화는 장기적인 안목에서 보면 웹 시스템 개발의 생산성 저하는 물론 사용자 요구사항과는 거리가 먼 웹 시스템을 개발할 가능성이 높고, 더 나아가 개발 비용의 초과 지출, 일정 지연, 낮은 품질 등 부정적인 결과를 낳을 수 있다.

1.2 웹 시스템 개발 프로세스 현황 및 문제점

본 논문에서는 현황 파악을 위해 다국적 IT 기업인 J사, Web Solution 전문업체인 C사와 국내 S사, Jim Conallen의

연구를 검토하였다. J사의 웹 개발 방법론은 (그림 1)에서 처럼 Discover, Design, Implement, Launch의 4개 단계와 이를 세분화한 15개의 Stage, 운영 및 유지보수를 위한 Ongoing 개발 프로세스, 이를 지원하는 기법을 정의한 techniques로 이루어져 있다[4].

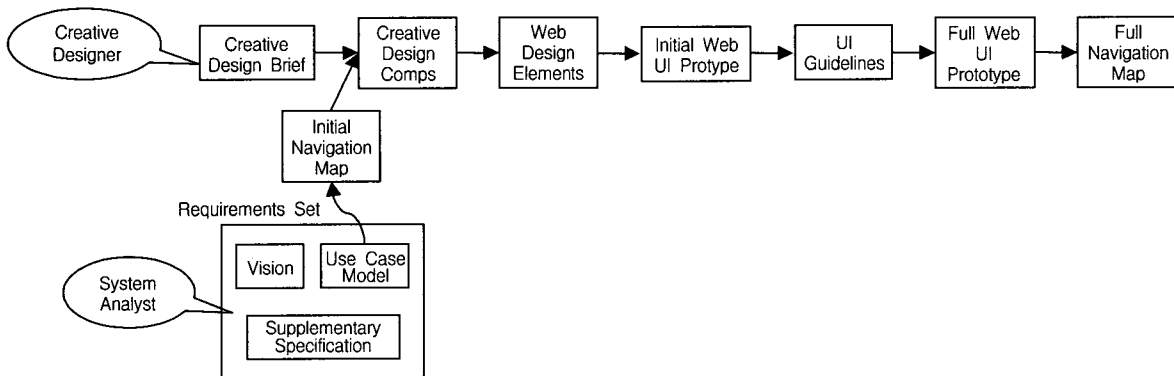


(그림 1) J사 웹 개발 방법론 구조

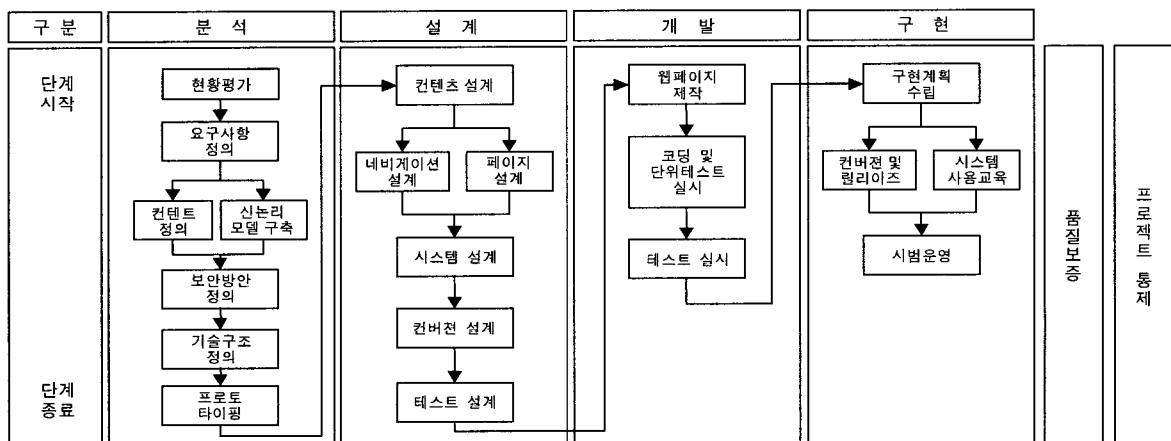
C사의 웹 개발 프로세스는 (그림 2)에서 보는 것처럼, 사용자 인터페이스와 프로토타이핑을 중심으로 프로세스가 구성되어 있고, 객체지향 중심으로 전개된다[5].

S사의 웹 개발 방법론은 프로젝트 착수, 분석, 설계, 개발, 구현, 프로젝트 종료의 6개 단계로 구성되어 있고, 관리 측면의 품질보증과 프로젝트 통제가 첨부되어 있다[6].

Jim Conallen의 프로세스는 웹 시스템 개발보다는 주로 소프트웨어 측면의 웹 어플리케이션을 개발하는데 초점을 맞



(그림 2) C사 웹 개발 프로세스 구조



(그림 3) S사 웹 개발 방법론 구조

추고 있으며, 객체지향적 관점에서 UML(Unified Modeling Language)을 웹 개발에 적용한다는 특징이 있다[7, 14]. 이 프로세스들을 분석해 볼 때, 다음과 같은 문제점들을 발견할 수 있다.

첫째, 개발 프로세스에 반복과 점증의 개념이 반영되어 있으나, 그 개념과 구조가 명확하지 않다. 웹 시스템 개발은 기술/관리적 측면에서 위험요소가 다수 잠재하고, 요구사항이 지속적으로 변경되는 특성이 있고, 신속한 유지보수를 특징으로 한다. 따라서 개발 프로세스에는 반복의 구성 및 반복별 필요 인원, 비용에 대한 고려가 포함되어야 한다.

둘째, 대상 도메인의 특성이나 비즈니스 영역을 고려하지 않은 일반적인 프로세스의 전개이다. 웹 기술은 구축되는 시스템의 유형과 규모에 따라 서로 다른 아키텍처와 프로세스를 나타낼 수 있다. 단순한 정보제공이나 홍보 목적의 사이트와 인터넷 또는 전자 상거래 사이트의 개발 프로세스는 규모 및 난이도에서 상당한 차이를 나타낼 것이므로 서로 다른 개발 프로세스가 적용되어야 바람직하다.

셋째, 프로세스 수행의 결과로 나오는 산출물이 제시되는 되어 있지만 그 흐름과 전개가 불명확하고, 개발자 또는 프로젝트 수행에 필수적 내용인지에 대한 고려가 부족하다. 일관성 있고 전후관계의 추적이 가능한 산출물 작성을 위해서는 통일된 접근방법과 표기법을 제시한 방법이 필요한데, 객체지향적 접근을 따를 경우, UML(Unified Modeling Language)이라는 모델링 언어에 근거한 표기법을 이용할 수 있다는 장점이 있다[8-10].

넷째, 전략에 대한 이해가 부족하다. 비즈니스 전략과 개발 전략을 정의한 웹 사이트 전략의 수립은 향후 프로젝트 전개에서 중요한 역할을 한다[11]. 그러나 전략에 대한 프로세스가 미흡하거나 제시되지 않아, 구현된 웹 사이트가 산만해지거나 일관성을 상실할 가능성이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 「도메인 특성에 근거한 웹 시스템 개발 프로세스」의 전제조건으로 도메인을 분류하는 방안과 특징을 정의하고 반복과 점증의 개념에 대한 전반적인 사항을 기술한다. 3장에서는 「도메인 특성에 근거한 웹 시스템 개발 프로세스」를 산출물명과 함께 제시한다. 4장에서는 인터넷 시스템을 사례로 들어 개발 프로세스의 적용상황을 살펴보고, 각 프로세스들에 대한 평가를 통해 타당성을 확인한다. 5장에서는 향후 개선사항을 제시하고, 결론을 맺고자 한다.

2. 도메인 특성에 근거한 웹 시스템 개발 프로세스의 요건

도메인을 분류[12, 13]하기 위해 웹 시스템을 구성하는 전체 요소들을 파악하고, 웹 사이트를 규모와 목적에 따라 구분한다. 분류된 웹 시스템 요소와 범주화된 웹 사이트간의 상관

관계를 맺어 분류된 영역들을 총 5개의 “도메인”으로 정의하였고, 각 도메인은 공통된 시스템 아키텍처와 유사한 규모 및 목적을 포함한다. 또한 위험관리에 바탕을 둔 반복과 점증에 대한 모형을 구체화하여 웹 개발 프로젝트 수행에 필요한 라이프사이클 모델을 제공한다.

2.1 도메인의 분류

2.1.1 웹 시스템 요소

기본적인 웹 시스템 요소는 클라이언트 브라우저, 네트워크, 웹 서버로 구성되며[14] 이 외에도 다양한 요소들이 유기적으로 결합되어 하나의 시스템을 이룬다. <표 1>은 웹 시스템의 구성요소들을 분류한 것이다.

<표 1> 웹 시스템 요소

대구분	소구분	구성요소 및 기능	대표적인 예
소프트웨어	사용자 서비스 영역	웹 브라우저	Explorer, Netscape
		객체 웹	ActiveX, Java Applet
		이메일	SMTP, POP3, IMAP4
		기타	Instance Messenger
	웹 서비스 영역	웹 서비스	HTTP server
		사용자 인터페이스	HTML
		비즈니스 로직	JSP, ASP
		데이터베이스 핸들링	JDBC, ADO
		트랜잭션 처리	JDBC, ADO
	비즈니스 서비스 영역	통신 매커니즘	HTTP
		비즈니스 로직	EJB, CORBA, DCOM
		데이터베이스 핸들링	EJB, JDBC
	데이터 서비스 영역	트랜잭션 처리	EJB
		통신 매커니즘	RMI/IIOP
		데이터베이스 관리시스템	Oracle, SQL-Server
운영체제 영역	비즈니스 로직	Trigger, Stored Procedure	
	트랜잭션 처리		
네트워크	유닉스	Unix	
	윈도우	Windows NT/2000	
하드웨어	통신 프로토콜	TCP/IP, RMI/IIOP, HTTP	
	서버 장비	-	
	클라이언트 장비	-	
	네트워크 장비	-	

이러한 구조는 기존의 클라이언트/서버 아키텍처를 근간으로 하고 있지만, 웹 고유의 요소들이 아키텍처 구성에 참여한다는 특징을 지니며, 이러한 요소들은 여러가지 형태로 서로 결합하고 연결되어 다양한 유형의 웹 시스템을 구성할 수 있다.

한 가지 고려할 것은, 최근 새롭게 접근되고 있는 솔루션 중심의 개발에서는 <표 1>의 내용들이 솔루션 자체에 구현되어 있으므로 분류된 내용들이 표면적으로 나타나지 않을 수 있다. 이 구성요소들 중 하드웨어 부문은 본 내용과 직접적인 연관이 없으므로, 이후에는 언급하지 않는다.

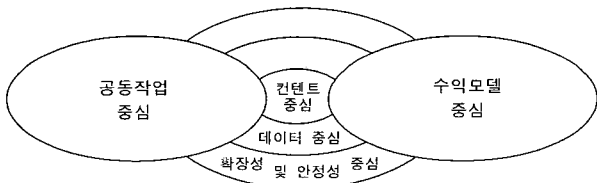
2.1.2 웹 사이트 형태 분류

웹 사이트는 대상 비즈니스 영역, 사용자 요구, 규모와 목적, 아키텍처의 구성에 따라 다양한 모습으로 나타난다. 요구되는 콘텐츠가 단순한 정보 전달에 있다면, HTML과 그래픽 요소가 결합된 웹 페이지로 충분하지만, 계속적인 유지보수와 시스템의 안정성이 중요하다면, 더 복잡한 형태의 아키텍처가 필요할 것이다[15]. 이러한 사항들을 고려하여 웹 사이트를 규모와 목적에 따라 <표 2>처럼 5개의 형태로 구분할 수 있다.

<표 2> 규모와 목적에 따른 웹 사이트 분류

구분	형태	정의	대표사이트 유형
규모에 따른 분류	콘텐츠 중심	단순 정보전달 목적의 웹 사이트 유형으로 HTML로 작성된 페이지 중심으로 구현되고, 데이터베이스는 사용되지 않거나, 소규모 사용에 그침	기업 홈페이지, 매뉴얼, 신문/광고
	데이터 중심	자료의 저장과 이용을 중요시하는 웹 사이트 유형으로 데이터베이스가 시스템에 통합된 형태를 보이며, 웹 페이지를 통한 단순한 자료의 조회부터 트랜잭션 처리까지 포함	소규모 인터넷, 커뮤니티, 예약 시스템, 은행, 우편주문
	확장성 및 안정성 중심	지속적인 유지보수와 안정성을 중요시하는 웹 사이트 유형으로 비교적 대규모의 형태를 보임	대규모 인터넷, 쇼핑몰, 포털, 경매
목적에 따른 분류	수익모델 중심	수익 및 매출과 연결되는 전자상거래가 주목적인 웹 사이트 유형으로 B2B가 주요 대상	각종 형태의 전자상거래
	공동작업 중심	기업체 또는 조직의 공동작업을 위한 웹 사이트 유형으로 비즈니스의 편리성과 신속성이 주요 목적	전자회의, 전자결재, 화상회의

(그림 4)는 위의 <표 2>에서 분류된 웹 사이트 유형의 포함 및 연관 관계를 그림으로 나타낸 것으로, 규모에 따라 분류된 콘텐츠, 데이터, 확장성 및 안정성 중심 웹 사이트의 포함관계와, 목적에 따라 분류된 공동작업 및 수익모델 중심 웹 사이트가 규모에 따라 분류된 3개 도메인의 영역 모두에 해당될 수 있음을 나타낸다.



(그림 4) 분류된 웹 사이트 유형간의 포함관계

2.1.3 웹 시스템 요소와 분류된 웹 사이트의 연결

웹 시스템의 요소들과, 분류된 웹 사이트들은 서로 연관성을 가지며, 이러한 관계 설정을 통해 웹 개발 프로세스의 전체적인 구조, 각 도메인별 개발 프로세스 및 중요 산출물이 정의될 수 있다. <표 3>은 하드웨어 부문을 제외한 웹 시

템의 요소와 분류된 웹 사이트의 관계를 친밀 정도에 따라 표시한 것이다.

<표 3> 웹 시스템 구성요소와 웹 사이트 유형의 상관관계

<●: 상, ◎: 중, ○: 하, *: 구원 아키텍처에 따라 다름, -: 관계 없음>

대구분	소구분	구성요소 및 기능	콘텐츠 중심	데이터 중심	확장성 및 안정성 중심	수익 모델 중심	공동 작업 중심
소프트웨어	사용자 서비스 영역	웹 브라우저	●	●	●	●	●
		객체 웹	◎	●	●	◎	◎
		이메일	●	○	○	●	●
		기타	○	●	○	○	●
	웹 서비스 영역	웹 서비스	●	●	●	●	●
		사용자인터페이스	●	◎	◎	●	●
		비즈니스 로직	●	●	*	*	*
		데이터베이스 핸들링	●	●	*	*	*
		트랜잭션 처리	●	●	*	*	*
	통신 매커니즘	●	●	●	*	*	
	비즈니스 서비스 영역	비즈니스 로직	-	-	●	*	*
		데이터베이스 핸들링	-	-	●	*	*
		트랜잭션 처리	-	-	●	*	*
		통신 매커니즘	-	-	●	*	*
	데이터 서비스 영역	데이터베이스 관리 시스템	◎	●	●	*	*
비즈니스 로직		◎	●	*	*	*	
트랜잭션 처리		◎	●	*	*	*	
운영체제 영역	유닉스	*	*	*	*	*	
	윈도우	*	*	*	*	*	
네트워크	통신 프로토콜	TCP/IP	○	○	◎	*	*
		RMI/IIOP	-	-	●	*	*
		HTTP	●	●	●	*	*

이러한 결과를 통해, 각 도메인별 기본 아키텍처를 구성해 볼 수 있고, 이를 중심으로 시스템을 구축하기 위해 필요한 프로세스의 방향을 예측할 수 있다.

가. 콘텐츠 중심 도메인

콘텐츠 중심 도메인의 주요 아키텍처는 클라이언트인 웹 브라우저와 웹 서버 중심의 구조를 나타낸다. 데이터베이스는 연동할 수 있으나, 그 역할은 크지 않고, 게시판, 회원관리 정도의 소규모 사용에 그친다. 개발 프로세스의 산출물에서 「웹 사이트 구조 설계」가 있다는 점이 대표적인 특징이고, 「콘텐츠 설계」와 「페이지 설계」가 중요한 역할을 한다.

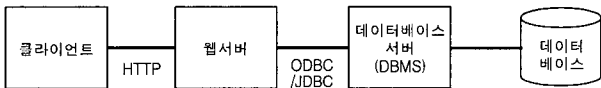


(그림 5) 콘텐츠 중심 도메인 아키텍처 모형

나. 데이터 중심 도메인

웹 브라우저, 웹 서버, 데이터베이스 서버 중심의 아키텍처

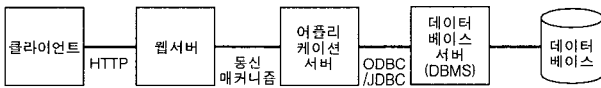
를 나타내며, 콘텐츠 중심 도메인과 차이점은 데이터베이스 측면의 역할이 강조되고 기업 내부의 기간 시스템과 연동하여 시스템을 구축한다는 점이다. 프로세스에서 「웹 사이트 구조 설계」보다는 외부시스템과의 인터페이스, 보안, 인프라에 대한 내용이 강조된다. 또한 프로세스와 데이터의 상관분석을 위한 「CRUD(Create/Read/Update/Delete) 매트릭스」를 적용하는데 이는 어플리케이션과 데이터베이스의 테이블과의 무결성을 검증하기 위해 사용한다.



(그림 6) 데이터 중심 도메인 아키텍처 모형

다. 확장성 및 안정성 중심 도메인

웹브라우저, 웹서버, 어플리케이션 서버, 데이터베이스 서버 중심의 아키텍처이며, 비즈니스 로직의 대부분을 어플리케이션 서버에 집중시킨다. 지역 단위 레벨의 복잡한 시스템 개발에 필요한 수준으로, 구체적인 도메인은 대규모 인터넷, 쇼핑몰, 포털, 경매 등 주로 분산형태를 취하며, 유지보수와 비즈니스 로직의 변경이 많고, 안정성이 중요한 사이트가 대상이다.



(그림 7) 확장성 및 안정성 중심 도메인 아키텍처 모형

라. 수익모델 중심 도메인

B2B, B2C 등의 다양한 전자상거래 사이트를 구축하는 것으로 목적에 따라 다양한 모델이 나올 수 있다.

마. 공동작업 중심 도메인

커뮤니티, 메시지, 정보교환 기능을 중심으로, 주로 기업체나 조직내의 공동 작업 또는 커뮤니케이션이 목적이다. 솔루션

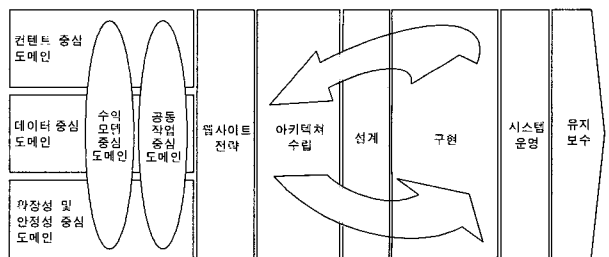
선 중심의 모델이 될 수도 있고, 해당 기업문화에 적합한 사용체계와 로직을 반영하는 것이 중요하다.

2.2 반복과 점증의 적용

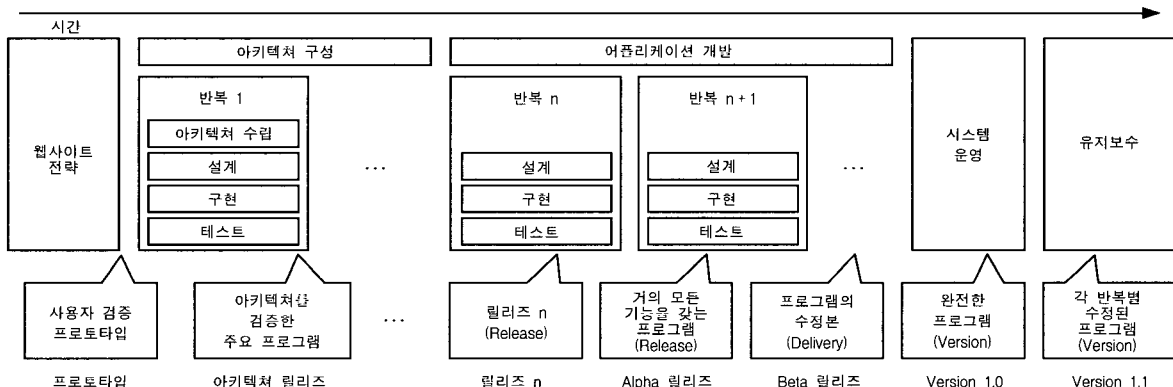
「도메인 특성에 근거한 웹 시스템 개발 프로세스」는 기본적으로 나선형(Spiral) 모델의 접근방법을 이용한다. (그림 8)은 반복과 점증을 적용하였을 때, 시간에 따른 반복의 유형을 살펴본 모형으로, 베이스라인[16]은 웹 사이트 전략, 아키텍처 구성, 어플리케이션 개발, 시스템 운영, 유지보수의 5개로 구분될 수 있다[5, 17, 18, 20].

(그림 8)처럼, 웹 사이트 전략에서는 비즈니스 전략 및 개발 전략을 포착하고 사용자 요구사항을 수렴하여 프로토타입을 개발하고, 아키텍처 구성에서는 웹 사이트 전략에서 표현된 유스케이스 중 위험도가 높은 부분을 중심으로 시스템의 아키텍처를 집중적으로 개발한다. 즉 사용자 인터페이스 측면보다는 아키텍처 검증에 중점을 두어, 반복이 끝나는 시점에서는 중요 항목에 대한 샘플 코드 및 설계 가이드 라인(아키텍처 명세서)이 나오는 것을 목표로 한다. 어플리케이션 개발에서는 프로토타이핑 및 해결된 아키텍처 요소를 중심으로, 지속적인 릴리즈를 통한 개발을 수행한다. 시스템 운영에서는 개발된 프로그램을 시스템에 로드하고 운영한다. 또한 사용자 매뉴얼을 작성하고 취합된 개선사항을 검토한다. 유지보수에서는 사용자측의 지속적인 피드백을 통해 웹 사이트를 관리, 평가, 갱신한다.

3. 도메인 특성에 근거한 웹 시스템 개발 프로세스



(그림 9) 도메인 특성에 근거한 웹 시스템 개발 프로세스 구조



(그림 8) 반복과 점증의 모형

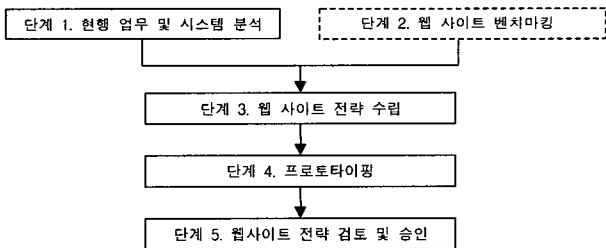
본 장에서는 5개로 분류된 도메인 중에서 콘텐츠 중심과 데이터 중심 도메인의 프로세스를 제시한다. 프로세스 위계는 “도메인 > 단계 > 작업”의 순서로 구성하며, 개발 웹 사이트의 특성에 따라 다른 도메인의 단계나 작업을 도입하여 커스터마이징할 수 있도록 한다. <표 4>는 콘텐츠 중심 도메인과 데이터 중심 도메인 프로세스의 차이점을 비교해 놓은 것이다.

<표 4> 콘텐츠 중심 도메인과 데이터 중심 도메인 비교

관 점	콘텐츠 중심 도메인	데이터 중심 도메인
도메인	<ul style="list-style-type: none"> 내용 전달 및 홍보 목적 콘텐츠 및 그래픽 요소 중심으로 주로 HTML 사용 데이터베이스 역할이 없거나 약함 웹 사이트의 전략 수립이 중요 	<ul style="list-style-type: none"> 기존에 구축된 시스템과 연동 데이터베이스를 중심으로 통합된 형태의 시스템 구축 시스템 구축을 위한 아키텍처가 중요
프로세스 및 산출물	<ul style="list-style-type: none"> 디자인 관점의 웹 사이트 구조설계, 콘텐츠 설계 및 페이지 설계가 중요 웹 사이트 기획자 및 디자이너의 역할이 중요 	<ul style="list-style-type: none"> 웹 사이트 구조 설계보다는 외부 시스템과의 인터페이스, 보안, 인프라에 대한 프로세스가 중요
대표 사이트	기업 홈페이지	중/소규모 인트라넷

3.1 웹 사이트 전략 워크플로우

「웹 사이트 전략」 워크플로우는 2개의 도메인에서 같은 경로로 수행된다. 현행업무 및 시스템 분석을 통해 요구사항을 정의하고, 벤치마킹을 통해 동종업체와 우수 웹 사이트를 비교 검토한다. 웹 사이트 전략을 구체화한 프로토타이핑을 통해 사용자 피드백과 승인을 얻는다.



(그림 10) 웹 사이트 전략 워크플로우

3.1.1 현행 업무 및 시스템 분석

대상 도메인에 현행업무 또는 시스템이 존재할 경우 수행하고, 발견된 문제점들은 요구사항의 기본 자료로 사용된다. 요구사항은 크게 기능적 요구사항과 비기능적 요구사항으로 구분하며, UML의 「유스케이스 다이어그램」으로 표현되고, 「유스케이스 보고서」를 통해 상세히 기술한다[8-10].

작 업	내 용	산 출 물
기초 자료수집	고객의 현재 업무와 시스템 현황을 파악하기 위한 자료 수집	용어집(Glossary) 자료수집 목록
현행 업무 분석	자료수집을 통해 얻어진 정보를 근거로 현재의 업무를 체계적으로 정리	현행업무 분석서 (액티비티 다이어그램)

작 업	내 용	산 출 물
현행 시스템 분석(선택)	자료수집을 통해 얻어진 정보를 근거로 현재의 시스템을 체계적으로 정리	현행시스템 분석서
기본 요구 사항 정의	위 3개의 작업에서 도출된 업무적, 기능적, 비기능적 요구사항 정의	요구사항 명세서 (유스케이스 다이어그램 및 유스케이스 보고서)

3.1.2 웹 사이트 벤치마킹(선택)

우수한 동종업체 또는 참고할 만한 타 웹 사이트의 현황을 파악한 후, 벤치마킹 업체 리스트, 벤치마킹 웹 사이트 구축 계획, 벤치마킹 웹 사이트 시스템 내역, 벤치마킹 웹 사이트 운영조직 및 환경, 벤치마킹 ROI(Return On Investment) 평가, 대상 웹 사이트의 목표, 기회, 차별화 전략 등의 기준이나 척도를 적용해 비교 검토한다.

작 업	내 용	산 출 물
동종업체 현황 파악	동종업체에서 웹 사이트를 운영하고 있는 현황 파악	벤치마킹 보고서
타 웹 사이트 평가	우수 웹 사이트를 선정하여 비교 검토 및 평가	

3.1.3 웹 사이트 전략 수립

업무분석과 시스템 분석의 결과를 바탕으로 구축하게 될 웹 사이트의 전략을 수립한다.

작 업	내 용	산 출 물
목표설정	웹 사이트가 지향하는 목표 설정 (목표/설명/영향사업분야/주요성공요인/우선순위)	전략수립 회의록 웹 사이트 전략(초안)
기회포착	웹 사이트 구축으로 얻을 수 있는 비즈니스 기회와 문제의 해결방안 정의	
차별화 전략 수립(선택)	웹 사이트를 차별화할 수 있는 전략 수립	전략수립 회의록 웹 사이트 전략(초안)

3.1.4 프로토타이핑

웹 사이트의 전략을 더욱 구체화하고 사용자측의 신속한 피드백을 받기 위한 프로토타이핑을 개발한다.

작 업	내 용	산 출 물
가상 시나리오 구성	구축될 웹 사이트의 모습을 가상 시나리오로 작성하여 프로토타입 영역 정의	프로토타입 시나리오
프로토타입 아키텍처 결정	프로토타이핑을 위한 아키텍처 정의	Action
프로토타입 설계	프로토타이핑을 위한 컨셉, 테마, 레이아웃, 네비게이션 구조 등에 대하여 개략적인 설계	
프로토타입 환경 구축	프로토타이핑을 위한 환경 설치	
프로토타입 사이트맵	프로토타이핑용 사이트맵 작성	프로토타입 사이트맵
프로토타입용 콘텐츠 수집	프로토타이핑 구축에 필요한 정보들을 수집하고 페이지 형식으로 변환	Action
프로토타입 구축	프로토타입 구현	구축된 프로토타입

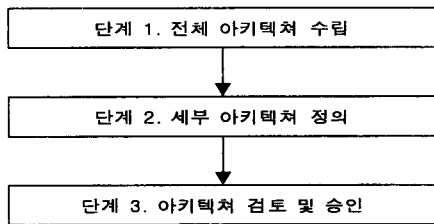
3.1.5 웹 사이트 전략 검토 및 승인

웹 사이트의 전략을 확정하고 사용자측의 승인을 얻으며, 다음 단계인 아키텍처 수립을 계획한다.

작업	내용	산출물
검토회의	웹 사이트 전략을 위한 검토회의를 개최하고 전략을 확정	Action
전략 승인	검토회의 결과를 반영하여 웹 사이트 전략을 수정	웹 사이트 전략(승인)
아키텍처 계획(선택)	확정된 전략을 바탕으로 아키텍처 단계 수행을 위한 프로젝트 계획 수립	프로젝트 계획서

3.2 아키텍처 수립 워크플로우

「아키텍처 수립」 워크플로우는 2개의 도메인에서 같은 경로로 수행되며 반복이 적용된다. 이 프로세스는 Top-Down 방식을 적용하여, 전체 아키텍처를 구성한 후, 세부 아키텍처를 정의한다. 세부 아키텍처의 요소는 페이지, 어플리케이션, 데이터베이스, 인터페이스, 인프라, 보안으로 구분한다.



(그림 11) 아키텍처 수립 워크플로우

3.2.1 전체 아키텍처 수립

전체적인 아키텍처에 대한 구조를 결정하고 이에 적합한 제품을 선정한다.

작업	내용	산출물
아키텍처 모델 선정	유스케이스 모델을 검토하고, 프로젝트 규모를 예측하여 확장성이 용이한 모델 선정	아키텍처 명세서
제품 선정	선정된 아키텍처에 맞는 제품들의 장단점 및 상호호환성을 고려하여 제품 선정	제품 선정안

3.2.2 세부 아키텍처 정의

전체 아키텍처 모델 및 제품을 기준으로 각 요소별 아키텍처의 세부사항을 결정한다.

작업	내용	산출물
페이지 아키텍처 정의	클라이언트 페이지에 포함될 콘텐츠 체계와 구조를 정의하고, 페이지의 논리적 사이트맵, 기술요소, 브라우저 등을 정의	아키텍처 명세서 사이트 맵 설계표준 정의서
어플리케이션 아키텍처 정의	클라이언트 페이지의 요청으로부터 해당 비즈니스 로직을 수행하는 방법을 결정하고 서버 스크립트 형식으로 로직을 구현	아키텍처 명세서 설계표준 정의서

작업	내용	산출물
데이터베이스 아키텍처 정의	주제영역을 선정하고 웹서버의 배치를 고려하여 데이터베이스 분산 여부를 결정	아키텍처 명세서 설계표준 정의서
기존 시스템과의 인터페이스 정의	인터페이스 대상을 조사하여 대상에 맞는 시스템 인터페이스 방법을 조사	아키텍처 명세서 설계표준 정의서
인프라 아키텍처 정의	하드웨어, 시스템 소프트웨어, 네트워크 아키텍처를 정의	디플로이먼트 다이어그램 아키텍처 명세서 설계표준 정의서
보안 아키텍처 정의	전체적인 보안정책을 수립하고 이를 시스템/네트워크/어플리케이션으로 세분하여 아키텍처 정의	아키텍처 명세서 설계표준 정의서
아키텍처별 매커니즘 정의	각 아키텍처별로 해결되어야 하는 문제점들을 나열하고 문제점들의 유형을 그룹핑	아키텍처 명세서

3.2.3 아키텍처 검토 및 승인

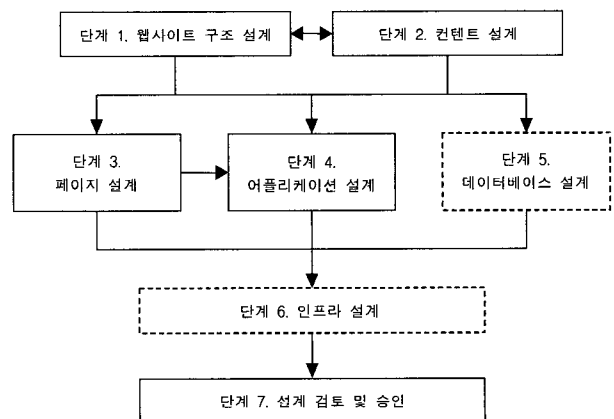
설계된 아키텍처를 검토하고 사용자의 승인을 얻는다.

작업	내용	산출물
검토회의	웹 사이트 아키텍처를 위한 검토회의 개최	Action
아키텍처 승인	검토회의 결과를 반영하여 아키텍처 수정	아키텍처 명세서(승인) 사이트 맵(승인)
설계 계획(선택)	확정된 아키텍처를 바탕으로 설계 단계 수행을 위한 프로젝트의 계획 수립	프로젝트 계획서

3.3 설계 워크플로우

「설계」 워크플로우는 2개의 도메인에서 각기 별도의 프로세스를 가지고 수행되고, 반복이 적용된다. 콘텐츠 중심 도메인은 웹 사이트 구조, 콘텐츠 및 페이지 설계에 초점을 맞추고, 데이터 중심 도메인은 웹 사이트 구조보다는 인터페이스, 보안, 인프라가 강조된다.

3.3.1 콘텐츠 중심 도메인



(그림 12) 콘텐츠 중심 도메인의 설계 워크플로우

가. 웹 사이트 구조 설계

페이지 디자인을 위해 컨셉, 테마, 레이아웃, 네비게이션, 레이블링 검색시스템의 구조를 설계한다.

작업	내용	산출물
컨셉 정의	웹 사이트의 주제를 선정하고 주제별 컨셉과 메인페이지 컨셉을 정의	웹 사이트 구조 설계서
테마 및 레이아웃 설계	테마와 레이아웃의 표준을 설계하고 이를 적용하기 위한 스타일 시트 작성	웹 사이트 구조 설계서 스타일 시트 페이지 설계서 페이지 Comp
네비게이션 구조 설계	웹 사이트 사용자가 이동 경로로 인식하게 되는 네비게이션 구조를 설계	웹 사이트 구조 설계서 사이트 맵
레이블링 구조 설계	정보를 효율적으로 전달하기 위해 레이블링 방법을 설계	웹 사이트 구조 설계서
검색 구조 설계	사이트 내에 검색엔진을 사용하기 위한 검색구조를 설계	웹 사이트 구조 설계서

나. 콘텐츠 설계

웹 사이트를 구성하는 내용이 되는 콘텐츠를 수집하고, 해당 페이지로 매핑하며, 페이지로 변환하기 위한 변환방법을 정의한다.

작업	내용	산출물
콘텐츠 수집 및 체계화	콘텐츠를 수집하고 체계화	콘텐츠 설계서
콘텐츠 매핑	분류된 콘텐츠를 매핑하고 표현하기 위한 페이지들을 선정하고 정의	콘텐츠 설계서
메인 페이지 콘텐츠 정의	메인 페이지의 콘텐츠를 정의하고 메인 페이지 구조 설계	페이지 설계서
콘텐츠 변환 방법 정의	콘텐츠를 웹 페이지에 넣기 위한 HTML 변환 방법을 정의	콘텐츠 설계서

다. 페이지 설계

각 페이지의 디자인을 위한 테마, 레이아웃, 그래픽을 설계한다.

작업	내용	산출물
테마 및 레이아웃 설계(선택)	각 페이지별 테마와 레이아웃을 설계하고 이를 적용하기 위한 스타일 시트 작성	스타일 시트 페이지 설계서
그래픽 설계	페이지에 포함되는 그래픽 요소를 구상하여 톨로 구현	Action
Comp 작성	페이지들의 기능적 특성과 그래픽 특성을 고려하여 Comp 작성	페이지 Comp

라. 어플리케이션 설계

비즈니스 로직이 필요한 동적 페이지를 선정하고, 이에 대한 어플리케이션을 설계한다.

작업	내용	산출물
동적 페이지 선정	사이트맵 및 페이지 설계에 나타난 페이지들에 대하여 비즈니스 로직이 필요한 페이지를 선정	Action

작업	내용	산출물
시나리오 작성	선정된 동적 페이지와 구현방법에 맞도록 페이지간의 네비게이션 및 어플리케이션 연동방법을 시나리오 형태로 설명	유스케이스별 시나리오 리스트(선택) 페이지 네비게이션 다이어그램
검토회의	어플리케이션 설계가 데이터 설계와 일관성있게 설계되는지 검증	페이지 네비게이션 다이어그램(확정)
어플리케이션 구조 결정	각 페이지의 시나리오에 참여하는 객체(페이지, 어플리케이션, 프레임 등)들의 구조를 결정하고 그 관계 표현	사이트 스트럭처 다이어그램
어플리케이션 명세	각 페이지 및 어플리케이션의 비즈니스 로직을 간략하게 정리	어플리케이션 명세서

마. 데이터베이스 설계(선택)

데이터베이스를 구축하기 위한 논리적인 데이터 설계와 이를 연계한 물리적 데이터베이스를 설계한다.

작업	내용	산출물
데이터 모델링	데이터의 구조를 모형화하기 위해 데이터 항목과 관계들의 구조를 논리적으로 표현하는 과정	ERD 테이블 명세서 엔티티 속성정의서
정규화	데이터베이스의 갱신요류를 피하고 중복저장을 최소화하며, 데이터 접근로직을 간단히 하기 위해 데이터 항목을 그룹화하고 분석하는 과정	ERD(수정) 테이블 명세서(수정) 엔티티 속성정의서(수정)
검토회의	데이터 설계가 어플리케이션 설계와 일관성있게 설계되었는지를 검증	CRUD(Create/Read/Update/Delete) 매트릭스 ERD(확정) 테이블 명세서(확정) 엔티티 속성정의서(확정)
데이터베이스 설계	데이터의 논리적 설계결과를 데이터베이스의 성능을 고려한 물리적 설계로 변환	ERD(물리)
스키마 생성	데이터베이스의 설계를 관계형 테이블로 변환하기 위한 SQL 정의 과정	DDL

바. 인프라 설계(선택)

하드웨어, 시스템 소프트웨어, 네트워크를 선정하고 설계한다.

작업	내용	산출물
하드웨어 선정	아키텍처 설계에 따라 필요한 하드웨어 구성요소와 사양을 파악하여 정의	시스템 명세서
시스템 소프트웨어 선정	아키텍처 설계에 따라 필요한 시스템 소프트웨어의 구성요소와 사양을 파악하여 정의	시스템 명세서
네트워크 설계	아키텍처 설계에 따라 필요한 네트워크의 구성요소와 사양을 정의	네트워크 및 시스템 구성도
인터넷 환경 설계	ISP 및 인터넷 연결방안의 정의	네트워크 및 시스템 구성도

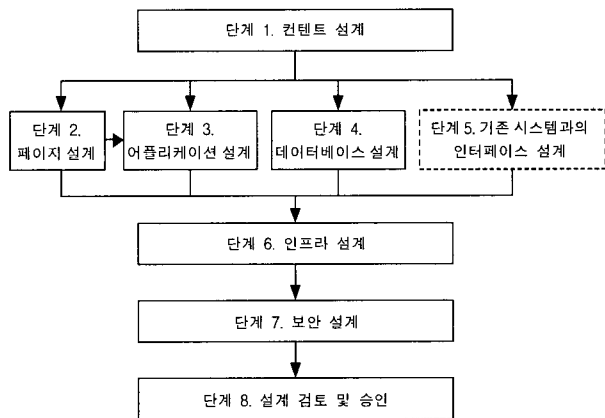
사. 설계 검토 및 승인

컨텐츠, 페이지, 어플리케이션, 데이터, 인프라를 검토하고

사용자측의 승인을 얻는다.

작업	내용	산출물
검토회의	설계 결과에 대한 검토회의를 개최하고 설계안을 확정	Action
설계승인	검토회의 결과를 반영하여 설계를 수정	웹 사이트 설계서
구현계획 (선택)	확정된 설계를 바탕으로 구현단계 수행을 위한 프로젝트의 계획 수립	프로젝트 계획서

3.3.2 데이터 중심 도메인



(그림 13) 데이터 중심 도메인의 설계 워크플로우

가. 콘텐츠 설계

웹 사이트를 구성하는 내용이 되는 콘텐츠를 수집하고, 해당 페이지로 매핑하며, 페이지로 변환하기 위한 변환방법을 정의한다.

작업	내용	산출물
콘텐츠 수집 및 체계화	콘텐츠를 수집하고 체계화	콘텐츠 목록
콘텐츠 매핑	분류된 콘텐츠를 표현하기 위한 페이지들을 선정하고 정의	콘텐츠 목록
콘텐츠 변환 방법 정의	콘텐츠를 웹 페이지에 넣기 위한 HTML 변환 방법을 정의	콘텐츠 목록

나. 페이지 설계

페이지의 디자인을 위해 컨셉, 테마, 레이아웃, 그래픽을 설계한다.

작업	내용	산출물
컨셉 정의	웹 사이트 전략을 토대로 전체 웹 사이트의 공통된 표준 컨셉을 정의	Action
테마 및 레이아웃 설계	테마와 레이아웃 표준을 설계하고 이를 적용하기 위한 스타일시트 작성	스타일시트 페이지 스케치
그래픽 설계	페이지에 포함되는 그래픽 요소를 구상하여 틀로 구현	Action
Comp 작성	페이지들의 기능적 특성과 그래픽 특성을 고려하여 Comp 작성	페이지 Comp

다. 어플리케이션 설계

콘텐츠 중심 도메인과 동일

라. 데이터베이스 설계

콘텐츠 중심 도메인과 동일

마. 기존 시스템과의 인터페이스 설계(선택)

기존 시스템이 존재할 경우, 연계 부분을 선정하고 인터페이스를 설계한다.

작업	내용	산출물
기존 시스템 연계 방법 설계	기존 시스템의 어플리케이션과 데이터베이스를 활용할 경우, 새로 개발되는 웹과의 연계 방법을 정의	시스템 연계 정의서
연계 구성	연계를 위한 구체적인 방안을 계획	시스템 연계 구성도 (선택)

바. 인프라 설계

콘텐츠 중심 도메인과 동일

사. 보안 설계

보안을 위한 방법을 정의하고 그에 따른 정책 및 절차를 설계한다.

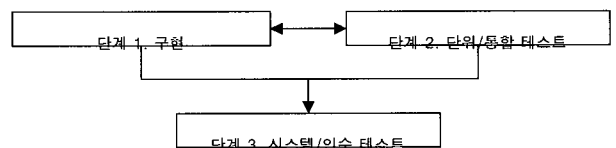
작업	내용	산출물
위험분석	위험분석 방법을 선정하고 위험요소를 파악 및 평가	Action
보안툴 선정 (선택)	보안을 수행할 보안 툴 선정	보안툴 선정안
보안 정책 및 절차 설계	아키텍처 설계단계에서 수립된 보안 정책에 따른 세부적인 표준절차나 통제 등을 설계	보안 정책 및 절차서

야. 설계검토 및 승인

콘텐츠 중심 도메인과 동일

3.4 구현 워크플로우

「구현」 워크플로우는 2개의 도메인에서 같은 경로로 수행되며 반복이 적용된다. 설계된 내용에 따라 인프라, 데이터베이스, 페이지, 어플리케이션, 보안 명세들을 구현하고, 단위, 통합, 시스템, 인수 테스트가 수행된다.



(그림 14) 구현 워크플로우

3.4.1 구현

설계된 내용에 맞게 시스템을 구현한다.

작업	내용	산출물
인프라 구축	설계된 사항을 바탕으로 기술설치 계획을 수립하고, 인프라 도입계약을 체결한 후 인프라 구축	인프라 계획서(선택) 계약서(선택) 구축된 인프라
데이터베이스 구축	설계 단계에서 설계한 물리 데이터베이스의 운영사항과 어플리케이션 지원사항을 정의하고, 트랜잭션 타임을 근거로 튜닝 작업을 수행한 후 데이터베이스 시험을 거쳐 물리 데이터베이스를 최적화시켜 구축	구축된 데이터베이스
페이지 개발	설계 단계에서 작성된 페이지 설계에 프로그래밍 요소를 추가하여 페이지 완성	개발된 페이지
어플리케이션 개발	설계 단계에서 작성된 설계에 따라 맞춤형 어플리케이션을 개발	개발된 어플리케이션
보안 구현	구축 및 개발된 인프라, 데이터베이스, 페이지, 어플리케이션을 설계된 보안 정책에 따라 구현	구현된 보안 솔루션 및 시스템

3.4.2 단위/통합 테스트

구현된 시스템이 기능적으로 완전한지를 테스트한다.

작업	내용	산출물
단위 테스트	구현된 요소들에 대한 단위테스트를 계획하고 테스트 수행	테스트 절차(선택) 단위테스트 데이터 정의서(선택) 단위테스트 결과 보고서
통합 테스트	구현된 요소들을 전체 통합하여 유기적으로 잘 작동하는지 테스트	통합테스트 계획(선택) 통합테스트 시나리오/데이터 정의서(선택) 통합테스트 결과 보고서

3.4.3 시스템/인수 테스트

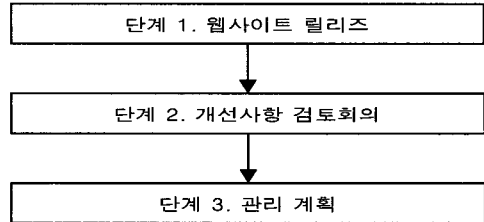
구현된 시스템 운영이 가능하며 사용자가 충분히 사용할 수 있는지를 테스트하고 승인한다.

작업	내용	산출물
시스템 테스트	시스템이 모든 기능 요구사항을 충족시키며 신뢰성있게 운영될 수 있는지 확인	시스템테스트 기준 정의서(선택) 시스템테스트 계획서(선택) 시스템테스트 참석자 명단(선택) 시스템테스트 결과 보고서
인수 테스트	사용자의 요구사항을 만족시키고 업무활동을 충분히 지원하는 것을 사용자로부터 확인	인수테스트 기준 정의서(선택) 인수테스트 계획서(선택) 인수테스트 참석자 명단(선택) 인수테스트 결과 보고서

3.5 시스템 운영 워크플로우

「시스템 운영」 워크플로우는 2개의 도메인에서 공통적으로 수행된다. 이 프로세스에서는 개발된 웹 사이트를 테스트 또는 라이브 컴퓨터(서버)에 로드하고, 사용자 교육을 실시한

다. 또한 사용자의 반응을 수집하여 개선을 위한 자료로 이용한다.



(그림 15) 시스템 운영 워크플로우

3.5.1 웹 사이트 릴리즈

웹 사이트 개통을 위한 사전준비와 웹 사이트의 릴리즈를 수행한다.

작업	내용	산출물
운영조직 구성 및 교육	웹 사이트 운영을 위한 조직을 구성하고, 교육 실시	운영자 매뉴얼
내부 홍보	사용자들에게 구축된 웹 사이트의 존재와 가치를 알리기 위해 홍보 활동을 수행	Action
클라이언트 지원	사용자들이 새로 구축된 웹을 원활하게 사용할 수 있도록 지원	사용자 매뉴얼
릴리즈 선언	웹 사이트의 접속 가능함을 사용자에게 공지	Action

3.5.2 개선사항 검토회의

사용자의 반응과 사용정도를 검토하여 웹 사이트의 개선에 반영한다.

작업	내용	산출물
사용자 반응 검토	릴리즈 이후 사용해본 사용자들의 반응을 수집하고 개선을 위한 자료로 활용	Action
사용도 검토	릴리즈 이후 모니터링을 통해 시간별, 사용자별 등 경영상의 필요한 기준에 따라 사용도를 조사	Action
검토회의	릴리즈 결과에 대한 검토회의를 개최하고 개선안을 확정	Action

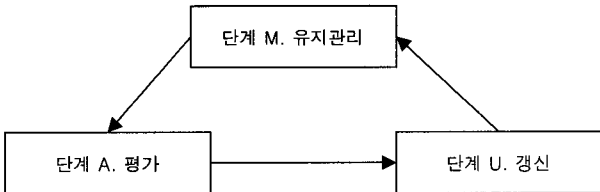
3.5.3 관리 계획

전체 시스템에 대한 최종 승인을 획득하고 이후 관리 및 유지보수에 대한 계획을 수립한다.

작업	내용	산출물
최종 승인	검토회의 결과를 반영하여 릴리즈를 수정	프로젝트 인수증
관리 계획	확정된 릴리즈를 바탕으로 관리단계 수행을 위한 프로젝트의 계획 수립	사후관리 계획서

3.6 유지보수 워크플로우

「유지보수」 워크플로우는 2개의 도메인에서 공통적으로 수행된다. 이 프로세스는 유지관리, 평가, 갱신이라는 단계들을 반복함으로써 사용자의 요구에 적합한 비즈니스 로직을 웹 사이트에 적용할 수 있도록 한다.



(그림 16) 유지보수 워크플로우

3.6.1 유지관리

컨텐츠와 시스템의 유지관리를 위한 여러 작업들을 수행한다.

작업	내용	산출물
컨텐츠 관리	컨텐츠의 품질, 컨텐츠 전달방법, 외부링크, 페이지타당성 등을 관리	Action
시스템 관리	시스템 무결성과 성능 최적화를 위해 시스템 관리	Action

3.6.2 평가

일정기간 웹 사이트에 대한 전반적인 실적을 검토 평가하여 웹 사이트 전략을 수정한다.

작업	내용	산출물
사용통계 검토	사용도의 개선을 위해 일정 기간 수집된 사용 통계를 모아 평가	평가서
접속유형 검토	모든 정보 영역이 실제로 유용한 정보로 설정되도록 하기 위해 일정 기간 수집된 접속 유형에 대한 자료를 모아 평가	평가서
응답시간 평가	사용자들이 보다 쉽게 사이트에 접속할 수 있도록 유도하기 위해 일정 기간 수집된 응답 시간에 대한 자료를 모아 평가	평가서
개선 요구 사항 검토	사용자들로부터 사이트의 개선 요구사항을 모아 분석	평가서
사용자 피드백 검토	개선의 결정을 할 때 반영하기 위한 사용자 피드백을 모아 검토	평가서
보안 이슈 검토	사이트에 분산되는 정보들이 안전한지 확인하기 위해 보안 이슈와 평가들을 모아 검토	평가서
업계 정보 검토	업계의 주요 정보를 검토하여 현 웹 사이트를 평가	평가서
신기술 검토	같은 업무를 보다 효과적이고 새로운 방법으로 수행할 수 있는 신기술 및 기술의 변화 검토	평가서
전략 수정	사이트의 최상위 목표와 정보 영역별 목표를 검토하고 보다 적합한 전략으로 수정	평가서

3.6.3 갱신

서비스와 텍스트 기반, 비주얼 기반의 컨텐츠를 갱신한다.

작업	내용	산출물
서비스 업그레이드	개선 요구사항에 설정된 우선순위에 따라 필요한 서비스 갱신	Action
컨텐츠 갱신	개선 요구사항에 설정된 우선순위에 따라 필요한 컨텐츠 갱신	Action

4. 적용사례 및 검증

4.1 적용사례

적용된 인터넷 시스템은 H사에서 2000년 7월부터 12월까지 지식공유 및 업무의 효율화 목적으로 설치 운영중인 웹 기반 인터넷 시스템으로, 내부 OA시스템, 지식관리시스템, 프로젝트 지원 등 총 5개의 시스템으로 구성되어 있다. 구축에 사용된 프로세스는 “데이터 중심 도메인”으로 Java 기반 기술인 EJB와 JSP, Applet 등을 사용하였고, 데이터베이스는 관계형 DBMS를 사용하였다.

4.1.1 웹 사이트 전략 워크플로우

공개가 제한되는 인터넷의 특성상, 타업체 인터넷 시스템의 상세한 구조와 내용에 대한 정보를 얻는 것은 어려웠기 때문에 벤치마킹 작업은 생략하였다. 시스템으로 구현할 주요기능은 이미 제시되어 있었지만, 더 정확한 요구사항의 파악을 위해 현행 업무의 내용과 흐름을 「현행업무 분석서」에 (그림 18)과 같이 “액티비티 다이어그램”으로 표현하였다.

여기서 도출된 문제점과 해결방안(그림 17)을 바탕으로 요구사항을 정의하였는데, 기능적 요구사항과 비기능적 요구사항으로 구분하여, (그림 19)과 같은「유스케이스 다이어그램」과 「유스케이스 보고서」에 정의하였다.

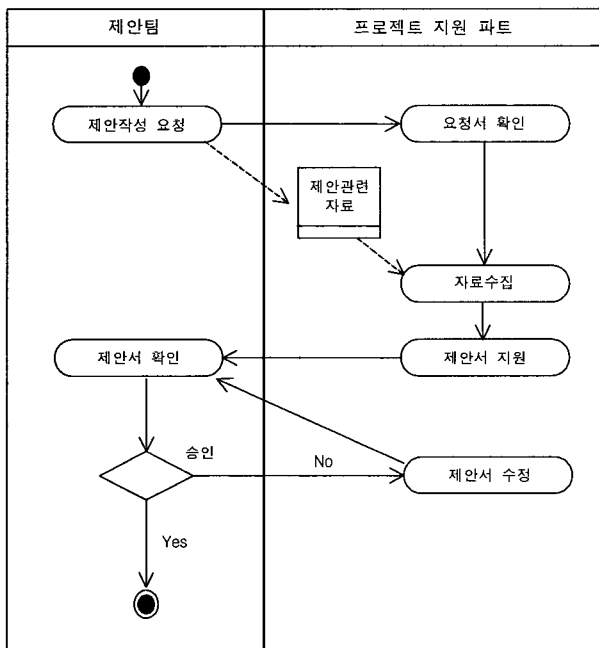
문제점	해결방안
• 방법론 지식을 효과적으로 프로젝트팀에 전달할 수 없음	• 체계화된 지원 시스템 개발
• 프로젝트팀에서 방법론을 적용할 시스템과 체계가 없음	• 방법론 customizing system 구축
• 프로젝트 수행 후 방법론 적용에 대한 결과 및 평가 누적이 없음	• 평가 시스템을 구축
• 프로젝트팀과 지원 팀간의 의사소통 창구가 없음	• 원활한 의사소통을 위한 지원 시스템 개발

(그림 17) 현행업무 분석서의 일부

이 요구사항을 바탕으로 웹 사이트의 목표 및 성공요인을 찾아내고 시스템으로 구현할 범위를 「웹 사이트 전략」을 통해 설정하였다. 요약된 웹사이트 전략의 목표는 다음과 같다.

- 효과적인 프로젝트 지원체계 수립
- 프로젝트팀과 지원파트간의 의사소통 개선
- 사내지식의 체계적 관리기반 마련
- 방법론/품질보증/신기술간의 통합 지원
- 사내거래제를 위한 기반 데이터 구축

또한 「웹 사이트 전략」의 내용들을 바탕으로 프로토타입 영역을 설정하고 구현하여 사용자들에 시연을 한 후, 업무흐름 및 추가 요구사항에 대한 피드백을 얻었다.



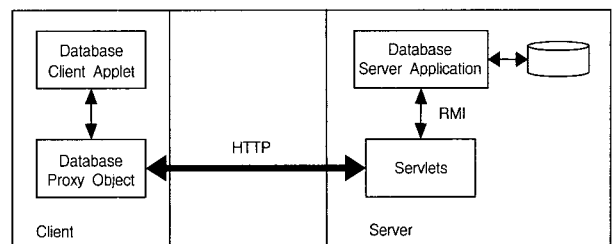
(그림 18) 현행 업무를 표현한 액티비티 다이어그램

의 구현에 중점을 두어 작업을 수행하였다. 개발자에게 산출물에 대한 부담감을 줄여 주기 위해 「아키텍처 명세서」와 「사이트맵」 2가지에 중점을 두었고, 세부 아키텍처 요소는 페이지, 어플리케이션, 데이터베이스, 기존 시스템과의 인터페이스, 보안, 인프라로 구분하였다.

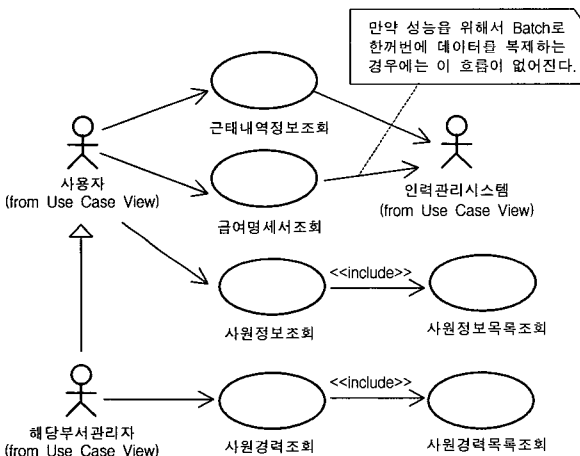
가장 위험도가 높고, 구현기간이 많이 소요되었던 요소는 트리구조의 애플릿, EJB와 관련된 비교적 신기술에 속한 부분이었는데, 일정이 1~2주 연기되는 원인이 되기도 하였다.

이 워크플로우의 종료시에는 위험이 내재되어 있는 아키텍처 요소들은 기술적으로 대부분 해결이 되었고, 「아키텍처 명세서」와 「사이트맵」을 통해 내용들을 표현하였다.

클라이언트 Applet을 위해 RMI(IIOP)트랜잭션을 수행하는 서버에서 수행되는 Java Servlet을 추가하여야 한다. 또한 HTTP로 Servlet과 통신하려면, 클라이언트 위에 RMI서버 객체의 method를 그대로 반영한 프락시 객체가 필요하다. (중략) 수정을 마치면 어플리케이션 아키텍처는 다음처럼 변해 있을 것이다 :



(그림 20) 아키텍처 명세서의 일부



(그림 19) 유스케이스 다이어그램

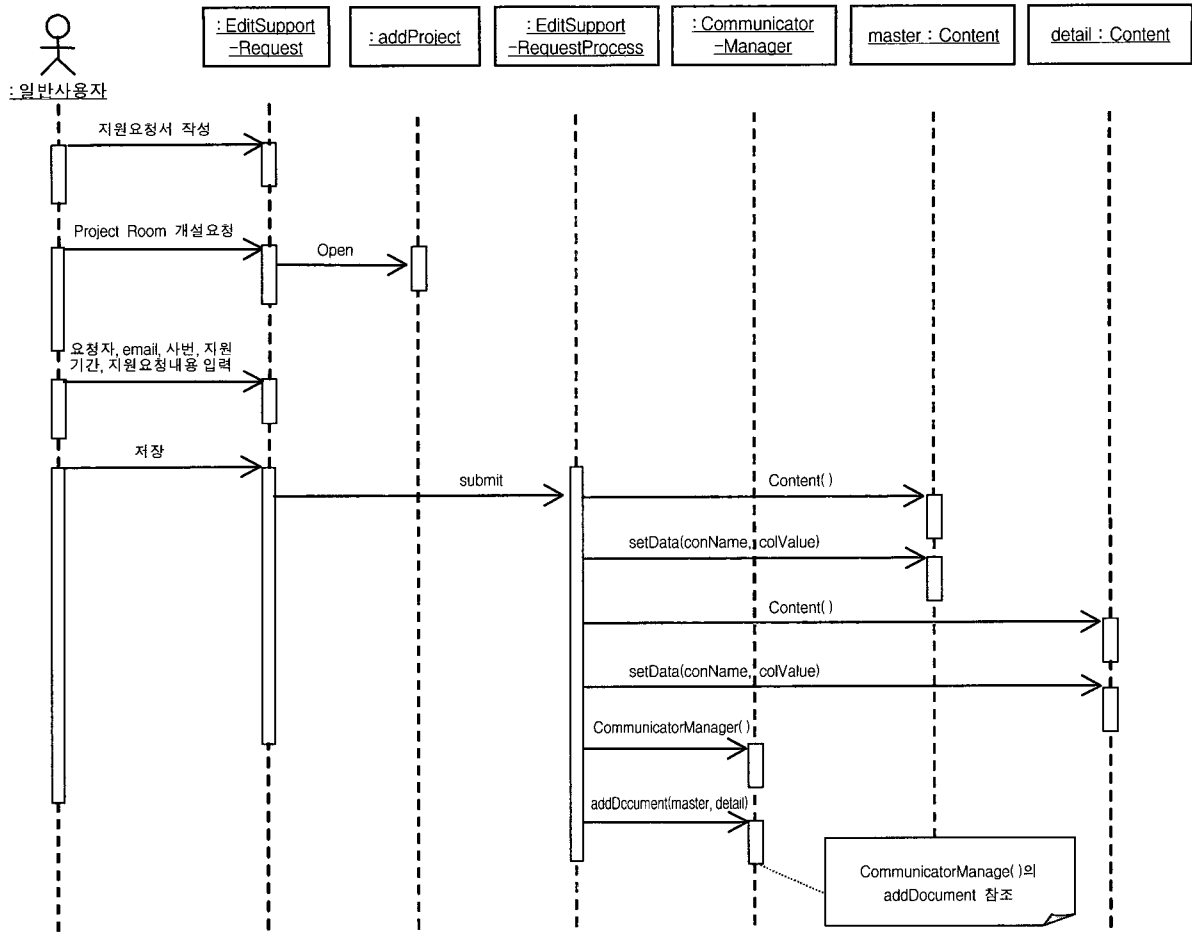
4.1.2 아키텍처 수립 워크플로우

아키텍처 수립 워크플로우는 위험도가 높은 주요 아키텍처

4.1.3 설계 워크플로우

설계 워크플로우는 아키텍처 수립 워크플로우의 요소인 콘텐츠/페이지, 어플리케이션, 데이터베이스, 인터페이스, 보안을 대상으로 수행하였다. 페이지 설계에서는 (그림 21)에서 처럼 네비게이션 체계를 UML의 “시퀀스 다이어그램”을 응용한 「페이지 네비게이션 다이어그램」으로 표현하였고, 어플리케이션 설계에서는 서버 페이지의 비즈니스 로직 및 관련 데이터베이스 내용을 DDL(데이터 정의어) 형식으로 정의하였다.

데이터베이스 설계에서는 개별 테이블과 각 테이블의 연결 상황을 나타낸 ERD(엔티티 관계도)로 나타냈고, 인터페이스는 인터넷 사용자들의 정보를 관리하기 위하여 기존 인사 시스템의 인사 데이터베이스로부터 사내 직원들의 정보를 비주기적으로 읽어오기 위한 로직을 설계하였다. 보안에서는 위험요소를 정의하고, 이를 근거로 논리/관리/물리로 분리하여 각각의 항목에 맞는 내용을 설계하였다.



(그림 21) 페이지 네비게이션 다이어그램의 일부

4.1.4 구현 워크플로우

구현 워크플로우는 아키텍처 수립 워크플로우에서 해결된 기술 요소들과 설계에서 작성된 내용들을 종합하여 시스템으로 개발하는 작업을 하였다. 이후 개발 내용에 대한 테스트를 계획하는 「테스트 계획서」를 정의하고 이에 대한 테스트 결과를 기록하였다.

4.1.5 시스템 운영 워크플로우

시스템 운영 워크플로우는 테스트까지 통과된 어플리케이션을 테스트 및 라이브 컴퓨터에 로딩하고 사용자 전체를 대상으로 웹 사이트 릴리즈에 대한 이메일 홍보와 브라우저 옵션조정에 대한 유의사항을 공지하는 작업을 수행하였다.

사용자 반응을 검토하기 위해 건의사항 및 수정 요구사항을 구현된 인트라넷 게시판에 올리거나 직접 운영자에게 이메일로 알려줄 것을 요청하였다. 또한 릴리즈 후, 간단한 매뉴얼을 작성하여 각 부서의 관리직원 또는 주사용자를 대상으로 오프라인 교육을 실시하여 인트라넷의 전체구조와 사용방법에 대한 내용을 전달하고 요구사항을 수렴하

였다.

4.1.6 유지보수 워크플로우

유지보수 워크플로우는 지속적인 관리와 평가, 콘텐츠 갱신에 대한 사항을 수행하는 단계인데, 담당부서의 요구와 일반 사용자들의 요청을 수렴하여 지속적인 보완활동을 해 나가고 있다.

4.2 개발 프로세스 평가

프로세스에 대한 평가는 “구성 측면”과 “요소 측면”으로 나누어 실시하였다. 구성 측면은 거시적인 관점에서 프로세스 전체의 성숙도에 중점을 두었고, 요소 측면은 미시적인 관점에서 웹 개발 프로세스에 포함되어야 할 항목들에 초점을 두었다[21-23].

4.2.1 프로세스 구성 평가

구성 평가는 프로세스 전체의 유기적 연결상황 및 전개의 합리성에 대한 성숙도를 평가한 것으로, 대상 항목은 대부분, 중구분, 소구분으로 세분화하고, 척도는 상, 중, 하로 구분하였다.

<표 5> 프로세스 구성 평가표

<●: 상, ◎: 중, ○: 하, -: 관계 없음>

대구분	중구분	소 구 분	J사	S사	도메인 기반
라이프 사이클 구조	단계 구분의 명확성	전략 수립	◎	○	●
		요구 분석	●	●	●
		설계	●	●	●
		구현	●	●	●
		테스트	●	●	●
		운영 및 유지보수	●	◎	●
	프로세스 모형의 성숙도	◎	○	●	
	프로세스 /데이터 상호관계	프로세스 지향	●	●	-
		데이터 지향	-	●	-
		객체지향	-	-	●
프로세스/데이터 지향접근법의 통합과 조화			◎	●	
프로세스 구조	개 요	프로세스 전체구조의 가시성	●	●	●
		단 계		●	●
	작 업	분석과 설계의 상호 연결성		●	●
		분석과 설계의 추적 가능성		●	●
		Task 구조의 조직화	●	●	●
		Task 완료에 대한 명확한 책임	●	●	●
		실행에 요구되는 Task 입력사항에 대한 명확한 정의	●	●	●
		작성되는 Task 산출물에 대한 명확한 정의	●	●	●
기법 및 가이드 제공	가이드의 상세도	●	●	◎	
	기법의 상세도	●	◎	◎	
	프로세스와의 조화도	●	◎	◎	
도메인 특성	아키텍처 구성 고려도	○	○	●	
	비즈니스 복잡도 고려도	○	○	●	
Case Tool	Case Tool 지원 가능성	◎	●	●	
	Case Tool과 프로세스의 적합성	○	●	●	
산출물	산출물 구성도 제공	○	●	●	
	구성요소 또는 템플릿 제공	◎	●	●	
결 과(총 27항목)					
구 분	관계없음	상	중	하	
J사	2	12	6	7	
S사	1	18	4	4	
도메인 기반	2	22	3	0	

4.2.2 프로세스 요소 평가

요소 평가는 웹 개발 프로세스에 필수적으로 포함되어야 할 내용들의 포함여부를 평가한 것으로, 대상 항목은 단계와 프로세스 요소로 세분화하였고, 척도는 유, 무로 구분하였다.

<표 6> 프로세스 요소 평가표

<√: 유, -: 무>

단계	프로세스 요소	J사	S사	데이터중심 도메인
전략 수립		-	-	√
요 구 분 석	벤치마킹	-	-	√
	현행업무 및 시스템 분석	√	√	√
	사용자 요구사항 파악	√	√	√
	프로토타입 구현	√	√	√
	아키텍처 구성	√	√	√
	제품 선정	√	-	√
	컨셉 및 테마 정의	√	√	√
설 계	레이아웃 작성	√	√	√
	네비게이션 체계	√	√	√
	페이지 구성 및 흐름	√	√	√
	서버 어플리케이션 설계	-	√	√
	데이터베이스 설계	-	√	√
	외부 인터페이스 설계	√	√	√
	그래픽 구성	√	√	√
	인프라 설계	√	√	√
	보안 설계	√	√	√
	구 현 및 테 스 트	페이지 구현	√	√
서버 어플리케이션 구현		-	√	√
데이터베이스 구현		-	√	√
외부 인터페이스 구현		√	√	√
그래픽 구현		√	√	√
인프라 구현		√	√	√
보안 구현		√	√	√
운 영 및 유 지 보 수	테스팅 수행	√	√	√
	사용자 지원 및 교육	√	√	√
	사용자 피드백	√	√	√
	운영 방안 및 조직구성	√	√	√
관 리	유지보수 방안	√	-	√
	검토회의 및 사용자 승인	√	√	√
	프로젝트 팀 구성	√	√	-
	구성원 역할	√	√	-
품질보증	√	√	√	
결 과 (총 33항목)				
해당 프로세스 요소 개수		27개	29개	31개
누락 프로세스 요소 개수		6개	4개	2개

4.2.3 프로세스 평가 결과

평가 결과, 3개의 프로세스는 <표 7>에서 보는 것처럼 다음과 같은 장점과 단점이 있는 것으로 분석되었다.

〈표 7〉 프로세스 평가결과

프로세스	장 점	단 점
J사	<ul style="list-style-type: none"> 프로젝트 수행시 구성할 조직 (웹 디자인, 기술, 콘텐츠 관리 등)에 대한 명확한 제시 사용자의 능동적인 참여와 역할에 대한 구체적인 작업 제시 컨셉, 테마, 그래픽 설계에 대한 구체적 내용 제시 	<ul style="list-style-type: none"> 전략에 대한 내용 미흡 반복과 점증에 대한 고려 부족 데이터베이스 설계 내용 미흡 프로세스 및 데이터의 상관관계 검증이 어려움 분석과 설계단계의 연결성 및 추적가능성 부족 단일 프로세스 제시로 인한 도메인 특성반영에 대한 유연성 부족
S사	<ul style="list-style-type: none"> 공정표, 공정도, 산출물 흐름도, 기법, Tool에 대한 명확한 구분 프로젝트 관리(통제, 착수 및 종료), 개발, 품질보증 프로세스에 대한 명확한 구분 수행방법과 고려사항을 통한 구체적 작업방향 제시 	<ul style="list-style-type: none"> 전략에 대한 내용 미흡 반복과 점증에 대한 고려 부족 운영 및 유지보수 측면 부족 단일 프로세스 제시로 인한 도메인 특성반영에 대한 유연성 부족
도메인 특성 근거	<ul style="list-style-type: none"> 구축할 도메인 특성과 비즈니스 영역에 근거한 프로세스 제시 반복 및 점증에 대한 개념을 근거로 구성하여 지속적 통합 가능 객체지향 접근 및 UML표기법의 사용으로 인한 산출물의 일관성 유지 웹 개발에 필수적 요소들을 반영한 개발자 중심의 프로세스 제시 	<ul style="list-style-type: none"> 프로세스 실현을 구체적으로 가이드할 수 있는 기법에 대한 설명 부족 기본적인 관리요소 중 팀구성 및 구성원 역할에 대한 내용 미흡

5. 결 론

본 연구는 웹 시스템 개발에 적합한 프로세스의 필요성을 제기하고, 현재 사용되고 있거나 공개된 웹 시스템 개발 프로세스를 통해 나타나는 문제점들을 지적하였고, 이를 해결하기 위해 「도메인 특성에 근거한 웹 시스템 개발 프로세스」를 제안하였는데, 이는 웹 시스템의 요소들과 규모 및 목적에 따라 분류된 웹 사이트의 상관관계를 통하여 그 방향을 설정하였다. 전제조건으로 반복과 점증, 위험관리를 위한 나선형 프로세스 모델을 선택하였고, 적용사례로는 인터넷 시스템을 사례로 들어 검증하였다. 또한 J사 및 S사의 프로세스와 도메인 기반 프로세스를 비교하기 위한 기준을 세우고 척도를 적용하여 평가하였다.

그 결과 기존의 웹 개발 프로세스는 구조적 기법 또는 정보공학 접근방법에 근거한 프로세스를 제시함으로써, 웹 개발처럼 사용자의 참여가 중요하고, 기술 및 관리 위험도가 높아 반복과 점증 기법이 필요한 상황에는 부적합한 것으로 판단되었다.

무엇보다 비즈니스 복잡도와 아키텍처 구성을 고려한 도메인의 특성을 고려하지 않아, 제공된 프로세스보다 규모가 크고 복잡한 구조의 프로젝트에 적용하려면 상당한 확대 커스터마이징이 따라야 한다는 점이 지적되었다. 개선된 의미로써 「도메인 특성에 근거한 웹 시스템 개발 프로세스」의 장점

은 다음과 같다.

- 웹 시스템 개발에 최적화된 전용 프로세스
- 도메인 규모와 아키텍처 특성을 고려한 프로세스
- 웹 사이트 전략을 통한 일관성있는 사이트 구축
- 반복과 점증을 통한 프로세스의 전개로 위험도 감소
- 초기의 프로토타이핑을 통한 사용자의 신속한 피드백 가능
- 프로세스 컴포넌트화로 필요시 개발 경로의 커스터마이징 가능
- 개발자에게 필수적인 산출물 제시

본 논문에서는 분류된 5개 도메인 중, 프로젝트에 적용된 2개의 도메인에 대한 프로세스를 제시하고 검토하였는데, 향후 나머지 도메인들에 대한 프로세스 개발이 필요하며, 특히 전자상거래 대상의 수익모델 중심 도메인은 산업 및 비즈니스 형태에 따라 다양한 프로세스의 개발이 예상된다.

또한, 단순히 프로세스 레벨에서 머무는 것이 아니라 프로젝트 관리요소와 개발기법 및 가이드 등을 충분히 보완한 “웹 시스템 개발 방법론”으로 정착시키는 작업과, 다양한 웹 개발 프로젝트에 적용하여 실제 프로젝트에서 발견된 문제점들을 근거로 프로세스를 수정하고 보완하는 피드백 작업도 큰 의미가 있을 것이다. 더 나아가 프로세스를 지원하는 케이스 툴과 상호연계하여 분석 및 설계의 모델링 작업에 일관성과 추적가능성을 제공하는 일도 고려되어야 할 것이다.

참 고 문 헌

- [1] 송관호, 윤태섭, 박종훈, 김형기, 신수정, “공공정보 시스템 환경 구축을 위한 기술 연구(웹)”, 한국전산원, 표준본부 기반 기술표준팀, pp.12-16, 1996.
- [2] 김영구 역, “웹 프로젝트 개발 가이드북”, 안그라픽스, p.3, 2000.
- [3] 최준용, 김영대, 서성채, 김병기, “웹 어플리케이션 특성에 관한 연구”, 정보처리학회 소프트웨어공학연구회지, 제3권 제4호, pp.74-75, 2000.
- [4] James Martin +Co, “WebWave,” 1998.
- [5] Rational Software Corporation, “Rational Unified Process 2001 A,” Rational Software Corporation, 2001.
- [6] 삼성SDS 품질경영팀, e~INNOVATOR, http://www.e-tqm.co.kr, 2000.
- [7] Jim Conallen, “Building Web Applications with UML,” Addison-Wesley, pp.3-4, 2000.
- [8] Grady Booch, James Rumbaugh, Ivar Jacobson, “Unified Modeling Language User Guide,” Addison-Wesley, 1999.
- [9] Martin Fowler and Kendall Scott, “UML Distilled 2nd Edition : A Brief Guide to the Standard Object Modeling Language,” Addison-Wesley, 2000.
- [10] 임춘봉, 신인철 공역, “UML 객체 모델링”, 인터비전, 2000.

[11] Vanessa Donnelly, "Designing Easy-to-use Websites," Addison-Wesley, pp.51-62, 2001.

[12] 문미경, 염근혁, "도메인 엔지니어링 관련 연구 동향", 한국정보과학회 소프트웨어공학회지 제14권 제2호, pp.5-6, 2001.

[13] 염근혁, "아키텍처 기반의 소프트웨어 개발을 위한 연구", 한국정보과학회 소프트웨어공학회지 제13권 제3호, pp.80-81, 2000.

[14] Jim Conallen, "Modeling Web Application Architectures with UML," Communications of the ACM, Vol.42, No.10, pp.63-64, Oct. 1999.

[15] 임현정, 박상용, "웹 정보시스템", 정일, pp.32-229, 2000.

[16] 이주현, "실용 프로젝트 관리론", 법영사, pp.312-314, 1994.

[17] Ivar Jacobson, Grady Booch, James Rumbaugh, "The Unified Software Development Process," pp.8-13, Addison-Wesley, 1999.

[18] 류형규, 이순천, 류시원, 신성호, "UML 기반 객체지향 클라이언트/서버 구축", 홍릉과학출판사, pp.81-87, 2000.

[19] 이호섭, "웹 기반 어플리케이션 설계 및 구현", 광운대학교, 전산대학원, 1999.

[20] 천유식, "소프트웨어 개발 방법론", 대청, pp.99-100, 1996.

[21] 송치양, 김혁만, 신의섭, 김수동, "전사적 통합 소프트웨어 개발방법론 구축", 정보과학회논문지(C), 제4권 제1호, pp.70-72, 1998.

[22] Deloitte Touche Tohmatsu, "4FRONT Methodology Manage4process Module," pp.Ⅱ-1-40-45, 1992.

[23] 정안나, 염근혁, "객체지향 소프트웨어 개발을 위한 품질보증 시스템 프레임워크", 정보과학회논문지(B) 제25권 제12호, pp.1775-1776, 1998.

[24] L. Baresi, F. Garzotto, P. Paolini, "Extending UML for Modeling Web Applications," Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001.



박수용

e-mail : sypark@ccs.sogang.ac.kr

1986년 서강대학교 전자계산학과 공학사
 1988년 Florida State University 전산학 석사
 1995년 George Mason University 정보기술학 박사, 연구 조교수

1996년~1998년 TRW Senior Software Engineer
 1998년~현재 서강대학교 컴퓨터학과 조교수
 관심분야 : 에이전트 지향의 소프트웨어 공학, 요구공학, 소프트웨어 아키텍처



임성재

e-mail : sjyim@hist.co.kr

1995년 서울대학교(학사)
 2001년 서강대학교 정보통신대학원(공학 석사)
 1996년~2001년 한진정보통신 기술연구소 주임연구원

2002년~현재 한진정보통신 경영지원본부 사업지원팀
 관심분야 : 소프트웨어 공학, 개발 방법론, 객체지향 분석 및 설계 (UML 기반)