ETRI

컴포넌트 기반 시스템 개발방법론 마르미-III 개발

2002. 1. 31.

ETRI
한국전자통신연구원

ETRI

목차

ETRI

마르미 History(1/3)

- 마르미 개발 방법론
  - 개발 동기
    - 개발업무의 체계 방법론 도입의 필요성
    - 외국 방법론의 도입 및 적용 등의 어려움
    - 국내 기술환경의 자체 개발의 필요성 대두
  - 개발 기간: 1994. 10. 1 ~ 1997. 9. 30.(3년간)
  - 구조적/정보공학 개발 방법론
  - 산학연의 공동 개발
  - 10개 업체에서 활용, 1개 업체 기술 이전 진행 중
  - 공공프로젝트에 적용
    - 표준정보서비스센터 시스템 개발에 일부 적용.

ETRI

마르미 History(2/3)

- 마르미-II 개발 방법론
  - 개발 기간: 1996. 11.1 ~ 1999. 10. 31.(3년간)
  - 주요 특징
    - UML 기반의 객체지향 개발 방법론
    - 유스케이스 기반
    - 아키텍처 중심
    - 반복/점진적 개발(Time box 개념)
    - 위험관리
    - 마르미 compliance
  - 2개 업체에서 활용, 1개 업체 기술 이전 진행 중

ETRI

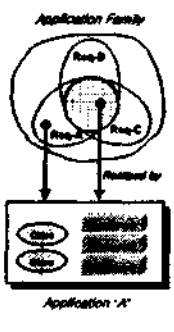
마르미 History(3/3)

- 마르미-III:
  - 컴포넌트 기반 개발 방법론의 필요성
    - 재사용성, 적시성, 유지 보수성 등이 업계의 공통적으로 대두
    - 그 실제적인 방법으로 제시되는 것이 컴포넌트 기반의 개발
    - [illegible]
  - 컴포넌트 기반 개발 방법론 프로토타입(FOCUS) 개발
    - 개발기간: 1999.8. ~ 2000.12.
    - 참여기관: 숭실대학교
  - 마르미-III 버전 1.0
    - 개발기간: 2001.1 ~ 2001.7.
    - 참여업체: [illegible]
    - KCSC 홈페이지(http://www.component.or.kr/)에서 다운로드 가능
  - 마르미-III 버전 2.0
    - 2002. 6. 예정
    - 이후 지속적인 업그레이드

ETRI

마르미-III의 특징(1/3)

- 재사용성 지향적
  - 한 도메인 내에서 같은 서비스를 제공하고 예견될 수 있는 차이를 가지고 있는 시스템의 집합(A set of items that have common aspects and predicted variability)
  - 공통성, 가변성 분석을 통한 컴포넌트의 재사용성 증대
- 개발 프로세스 분리 정의
  - 컴포넌트 개발 프로세스(CDM)
  - 컴포넌트 기반 어플리케이션 개발 프로세스(CBD)
- UML 사용
  - 1997. 11., OMG에서 UML을 객체 지향 모형화 언어 표준으로 채택
  - Process Independent
  - Tool Independent

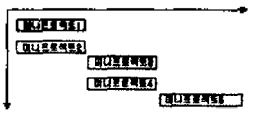


마르미-III의 특징(2/3)

- Use case 기반
  - 사용자 요구사항 분석의 중심 기법
  - 컴포넌트 식별 시 고려
  - 점진적 개발 단위
- EJB 명세 기반
  - 구현 플랫폼 대상 Enterprise JavaBeans
- 아키텍처 중심
  - 시스템 품질을 고려한 재사용성의 증대
  - 개발 초기에 견고한 아키텍처 정의
  - 마르미-III의 시스템 아키텍처
    - 비즈니스 컴포넌트 아키텍처
    - 응용 컴포넌트 아키텍처

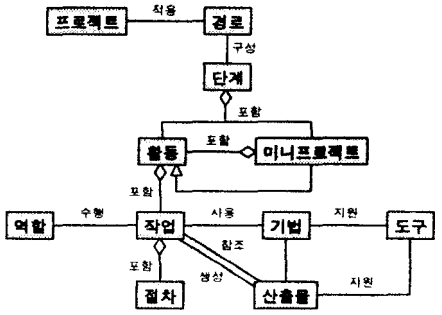
마르미-III의 특징(3/3)

- 반복적, 점진적 개발
  - 미니프로젝트(Time Box)
    - 소규모팀, 집중된 개발 일정
    - 1개 이상 컴포넌트 개발
    - 순차적, 병렬적 진행
  - 고객중심의 변경 관리
  - 위험관리
  - 개발자의 지속적 동기 부여



- 위험관리
  - 초기에 위험의 규명 / 평가 / 해결 방안 마련(위험분석서)
  - 체계적이고 지속적인 위험 관리
- 마르미, 마르미-II와의 일관성
  - 기존 마르미, 마르미-II 사용자의 적응성 제고
  - 방법론 메타모델, 관리 및 지원 툴등과 같은 공통 부분 활용

방법론 메타 모형(1/2)

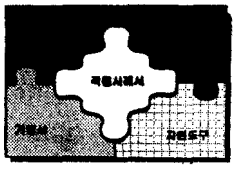


방법론 메타 모형(2/2)

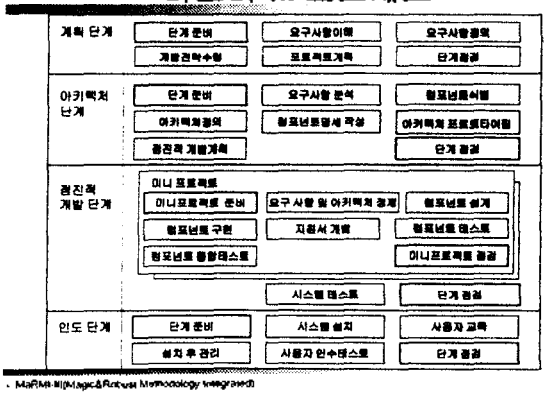
- 단계(phase)
  - 방법론 구성 작업 단위의 최상위 수준으로 활동의 구조적 집합
- 활동(activity)
  - 논리적 연결성이 있는 작업의 구조적 집합
  - 모든 활동에는 점검 사항을 계시
- 작업(task)
  - 개발자가 수행하는 최소 단위의 일로 하나 이상의 절차로 구성
  - 단일 목표를 성취하기 위한 과정으로 프로젝트 계획 수립의 기본 단위
- 절차(procedure)
  - 방법론 계층의 최하위 수준으로 작업을 수행하기 위한 문서에 해당
- 미니프로젝트(mini project)
  - 활동의 한 유형으로 제한된 시간 내에 사용 가능한 시스템을 생성
- 기법(technique)
  - 전문적인 절차와 개념, 기술을 사용하여 작업을 완수할 수 있는 수행방법
- 도구(tool)
  - 작업을 효율적으로 수행하기 위한 수단
- 역할(role)
  - 프로젝트 내에서 작업을 수행하는 조직 또는 사람
- 산출물(artifact)
  - 작업의 수행결과로 생성되는 문서나 파일

마르미-III v1.0 구성

- 절차서
  - 계획단계
  - 아키텍처 단계
  - 점진적 개발 단계
  - 인도단계
- 기법서
- 양식정의서
- 적용사례서
  - 체육관리시스템



마르미-III 프로세스



계획단계

■ 특징

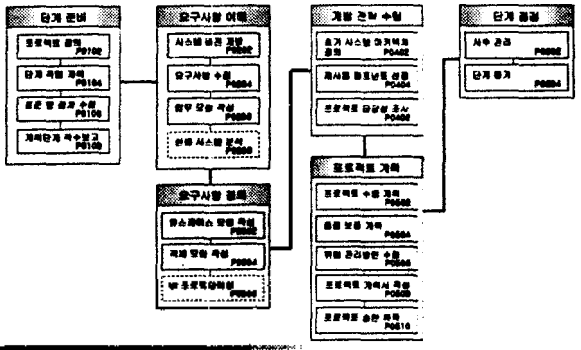
◆ 개발 시스템의 배경, 목표, 및 범위를 결정한다.

◆ 사용자 요구사항 및 관련 자료를 수집한 후, 유스케이스 및 객체 모형의 작성을 통해 시스템의 범위를 정의하고 프로토타이핑을 통한 사용자의 검증을 수행한다.

◆ 작성된 모형을 기반으로 전체 프로젝트의 규모 및 소요비용 추정하고 타당성 분석을 수행한다.

◆ 시스템을 개발하기 위한 초안 계획 및 품질보증계획을 수립하여 프로젝트계획서를 작성하고 사용자와 의사결정자에게 승인을 받는다.

계획단계 구성도(CBD)



계획단계 주요 활동

■ 유스케이스 중심의 요구 사항 파악

◆ 유스케이스를 중심한 사용자 중심의 요구사항 파악 및 관리
◆ 사용자에게 친숙한 용어와 시나리오를 활용한 요구사항 파악
◆ 개념적 객체 모형 작성
◆ UI 프로토타이핑

■ 개발 전략 수립

◆ 초기 시스템 아키텍처 정의
- 유사한 시스템 또는 과거의 문제점부터 파악한 경험을 기반으로 도출한 아키텍처
◆ 재사용 컴포넌트 후보 파악
- 초기 아키텍처와 유스케이스 및 개념모형을 기반으로
◆ 시스템의 개발 배경과 효과를 분석
◆ 프로젝트의 타당성 검토

■ 체계적인 프로젝트 계획 수립

◆ 수행계획, 관리계획, 품질 향상 계획을 포함하는 포괄적인 프로젝트 계획 수립

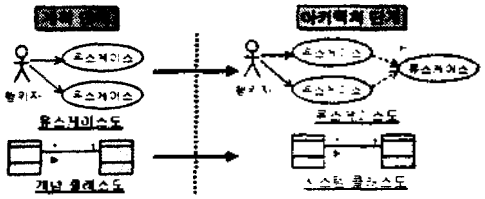
유스케이스 중심의 요구사항 파악

■ 유스케이스

◆ 사용자가 목적하는 시스템의 기능을(functionality)
◆ 요구 추적, 요구 관리, 개발, 테스트의 주요 단계로 이용
◆ 계획단계로 정립단계에서 업무되며 사용자와 함께 요구사항을 분석

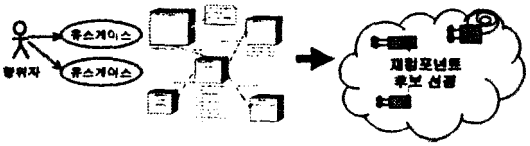
■ 계획과 아키텍처 단계의 요구 분석 산출물의 구분

◆ 계획 단계의 아키텍처 단계 산출물의 작성 내용 구분



개발 전략 수립

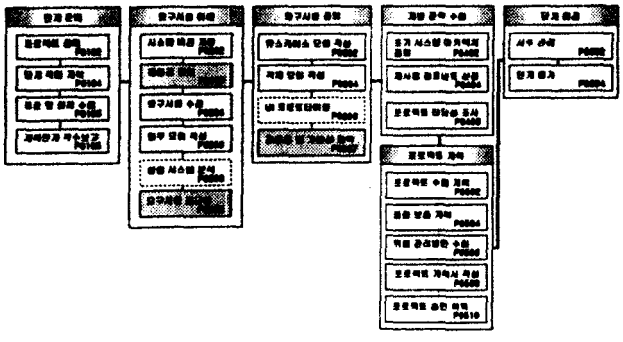
■ 초기 아키텍처를 바탕으로 재사용 컴포넌트 조사



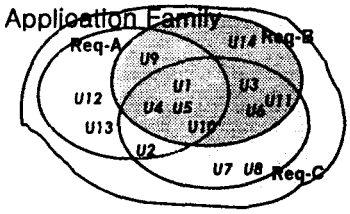
■ 프로젝트 타당성 분석

◆ 규모/비용 견적
◆ 투자 효과, 상품성 분석

계획단계 구성도(CDM)



공통 유스케이스 파악



가변성(1/2)

▣ 정의

- [illegible]

▣ 가변성 요인

- [illegible]

가변성(2/2)

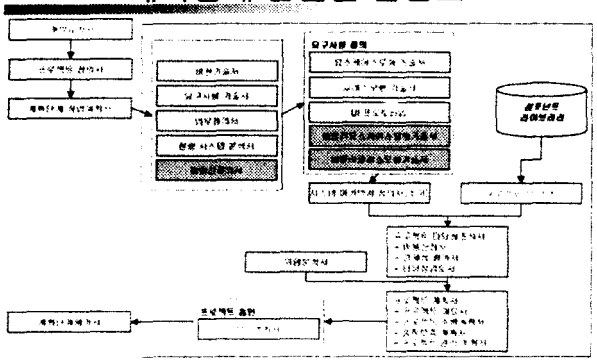
1. 공통성 및 가변성 테이블

유스케이스명		멤버1명	멤버2명	멤버n명	공통성	가변성
UC-1	속성	Customer, Order, Credit	Customer, Order, Cash	...	Customer, Order	Credit, Cash
	기능					
UC-2	속성					
	기능					
공통성 정도						

2. 가변성 범위 테이블

유스케이스명	가변요소명	예상여부	범위	초기값
UC-1	makePayment	No	(Cash, Card, e-cash)	Card

계획단계 산출물 연관도

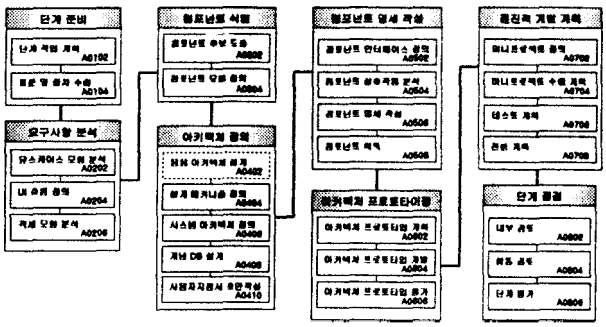


아키텍처 단계

▣ 목적

- 요구사항 정련을 통해 아키텍처 중심 요건을 파악하고, 이를 토대로 구현 가능한 응용 아키텍처와 재사용성 높은 비즈니스 아키텍처를 포함하는 **시스템의 아키텍처를 결정**한다.
- 다양한 기법을 통해 비즈니스 컴포넌트를 식별하고, 응용 아키텍처를 지원하는 응용 컴포넌트를 결정한 후 최종 컴포넌트 명세를 작성하여 **컴포넌트를 도출**한다.
- 아키텍처 프로토타입을 통해 아키텍처적인 위험요소를 발견하고, 이를 반영하는 시스템 아키텍처를 결정하여 향후 구현 과정에서 발생 가능한 **아키텍처 위험요소를 제거**한다.
- 점진적인 시스템을 개발하기 위한 계획을 수립한다.

아키텍처 단계 구성도(CBD)



아키텍처 단계 주요 활동

(가) 요구사항 분석
- 가변성을 고려한 일반화 관계(Generalization)와 포함(Include), 확장(Extend), 구현(Realize) 관계를 정의하고 공통 유스케이스를 도출하여 유스케이스를 구조화(정제)함.
- 계속 적재 모델을 사스템 적재 모델로 상세화

(나) 컴포넌트 식별 및 아키텍처 정의
- 유스케이스와 객체 모델을 바탕으로 컴포넌트 식별
- 컴포넌트 인터페이스 정의
- 컴포넌트 realization 정의

(다) 점증적 개발을 위한 개발 계획 수립
- 순차적/병렬적 개발을 고려한 MILI프로젝트 정의 및 수행 계획
- 컴포넌트 테스트 및 통합테스트 등을 테스트 계획 수립

컴포넌트 식별 방법

(가) 4가지 유형에 대한 지침 제공
- 객체모델을 통한 컴포넌트 도출
- 공통 유스케이스를 통한 컴포넌트 도출
- UI프로토타입을 통한 컴포넌트 도출
- UDA(Use case Data Access)를 통한 컴포넌트 도출

UDA를 활용한 컴포넌트 식별(1/4)

(가) Use case-Class 관계를 이용한 클러스터링 방법
- 컴포넌트에 클래스 이상의 업무적 관점의 반영
- Mechanical Approach

(나) Steps:
- 유스케이스 모델링
- 객체모델링
- Use case-Class 관계표 작성

	C1	C2	C3	C4	C5	C6
U1	C	R	W			R
U2	R	R	C	C		
U3	R	C				C
U4	R			R	C	
U5				R	D	
U6			R	R		
U7	R	R			R	R
U8	R	R	W	D		

UDA를 활용한 컴포넌트 식별(2/4)

(가) Use case-Class 관계표 조정
- 클래스를 참조(read)만 하는 유스케이스의 추출
- 공통 유스케이스 추출
- 공통 클래스 추출
  - 공통 유스케이스와 생성, 수정, 삭제 관계가 있는 클래스
  - Exclusive --> 공통 컴포넌트
  - Non-exclusive

Base class(=exclusive)

Base use cases

Referencing only use cases

UDA를 활용한 컴포넌트 식별(3/4)

(다) 관련도 계산

A(C) = 10, A(R) = 7, A(W) = 5, A(D) = 3 이라고 하면,

$R(U_i, C_j) = A(U_i, C_j) / \sum_{k=1 \sim n} A(U_i, C_k)$

여기서, $1 \le i \le n$, $1 \le j \le m$

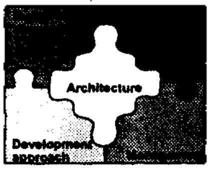
UDA를 활용한 컴포넌트 식별(4/4)

(라) 후보 컴포넌트 식별
- 임계값: 0.35
- 클러스터링

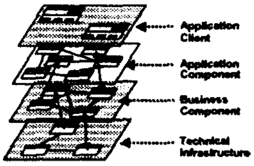
ETRI

시스템 아키텍처

- 모든 실제적인 재사용은 아키텍처를 중심으로 가능함.
- 개발 초기에 아키텍처를 결정하고, 이를 중심으로 개발 프로세스, 관리, 비즈니스 변경 요구의 수용과 지속적인 재사용 구조를 구축해 나감.
- 대부분의 잘 구성된 아키텍처는 경험을 통해 만들어 짐.(non-deterministic)
- 시스템 아키텍처 = 비즈니스 컴포넌트 아키텍처 + 응용 컴포넌트 아키텍처



The architecture is the foundation of everything.

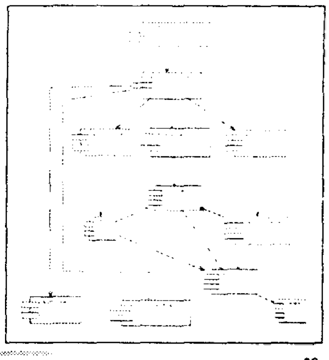


Component System Architecture

ETRI

비즈니스 컴포넌트 아키텍처

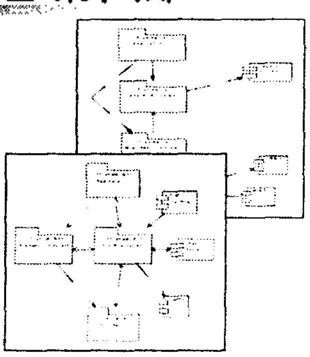
- 정의
  - 유스케이스와 시스템 객체를 바탕으로 식별된 비즈니스 컴포넌트를 토대로 컴포넌트간의 관련성을 고려하여 구조화한 아키텍처
- 이 컴포넌트 모형은 컴포넌트 식별 활동에서 작성 되고, 아키텍처 정의 활동에서 정제를 거쳐 최종 컴포넌트 아키텍처 모형으로 작성됨



ETRI

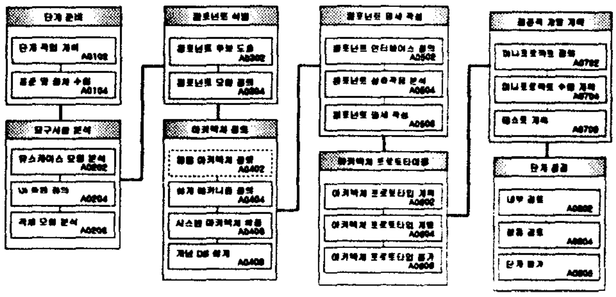
응용 컴포넌트 아키텍처

- 정의
  - 응용 컴포넌트는 비즈니스 컴포넌트와 구현을 위해 필요한 기술적인 컴포넌트
  - 그리고 응용 컴포넌트는 프레임워크 기술을 이용하여 구현하게 되며 기술 아키텍처 측면에서 비즈니스 컴포넌트의 수행을 지원하는 역할을 수행함
- 순서
  - 기술 요구사항
  - 아키텍처 요구사항 및 제약사항
  - 아키텍처 평가기준
  - 기존 아키텍처 정의
  - 기존 아키텍처를 평가
  - 초기 응용 아키텍처 정의



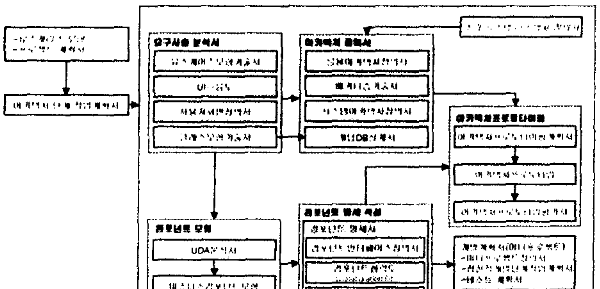
ETRI

아키텍처 단계 구성도(CDM)



ETRI

아키텍처 단계 산출물 연관도

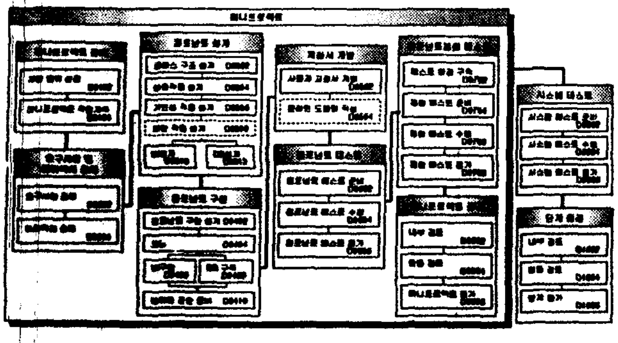


ETRI

점진적 개발 단계

- 목적
  - 개발 시스템을 유스케이스를 기준으로 미니프로젝트 단위로 분할하여 개발한다.
  - 미니프로젝트별로 구축할 유스케이스를 구현 관점에서 보완하고, 이를 바탕으로 아키텍처를 또한 구현 관점에서 보완한다.
  - 식별된 컴포넌트의 내부를 설계하고, 데이터베이스, 사용자 인터페이스 등을 설계하고 구축한다.
  - 개발된 컴포넌트에 대해 인터페이스가 제대로 구현되었는지 검사하기위해 컴포넌트 테스트를 수행하고, 컴포넌트간 통합 테스트를 수행하면서 통합한다.
  - 각 미니프로젝트가 끝난 후, 개발된 시스템의 기능성 및 성능이 사용자의 요구를 만족하는지 확인하기 위해 시스템테스트를 실시한다.

점진적 개발 단계 구성도(CBD)



MaRMI-III(Magic&Robust Methodology Integrated) 37

점진적 개발 단계 주요 활동

- 점진적 개발
  - 유스케이스 정제(구현 관점)
  - 컴포넌트(DB, UI 포함) 상세 설계 및 구현
  - 일련의 미니프로젝트 수행을 통하여 시스템 구축
- 테스트 설계 및 수행
  - 컴포넌트의 기능성을 테스트하기 위한 컴포넌트 테스트 설계 및 수행
  - 컴포넌트 사이의 통합테스트 설계 및 수행
  - 미니프로젝트마다 시스템 테스트 설계 및 수행

MaRMI-III(Magic&Robust Methodology Integrated) 38

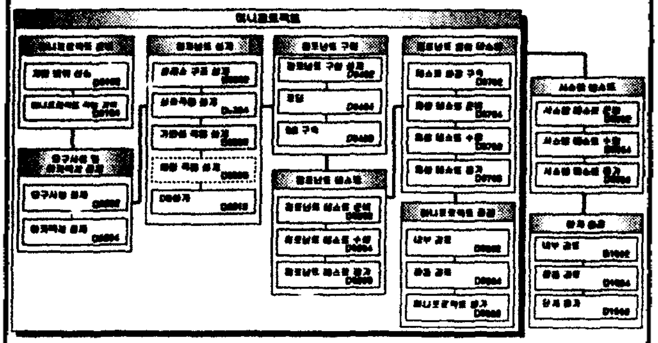
미니 프로젝트

- Time Box 개념
  - 소규모 팀(1~4명)
  - 짧은 개발 일정(2~6주)
  - 1~3개의 비즈니스 컴포넌트와 연관된 유스케이스(집합)를 개발
- 점증/순차적, 동시/병렬/분산 개발 또는 이의 혼합이 가능
- 사용자는 시스템이 개발되는 과정과 결과를 확인
- 고객중심의 변경관리 용이
- 주기적인 위험 관리 용이
- 개발자는 조기에 피드백을 받을 수 있고, 주기적인 동기 부여로 생산성 향상 도모

시스템 개발 과정

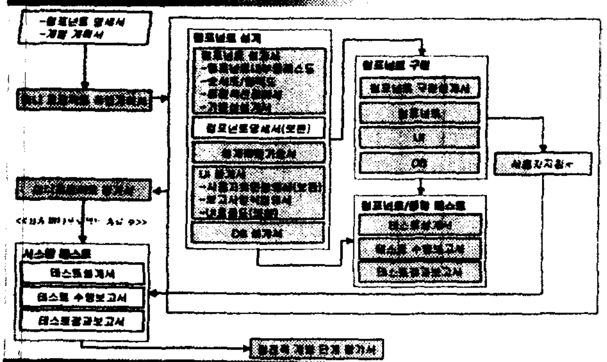
MaRMI-III(Magic&Robust Methodology Integrated) 39

점진적 개발 단계 구성도(CDM)



MaRMI-III(Magic&Robust Methodology Integrated) 40

점진적 개발 단계 산출물 연관도



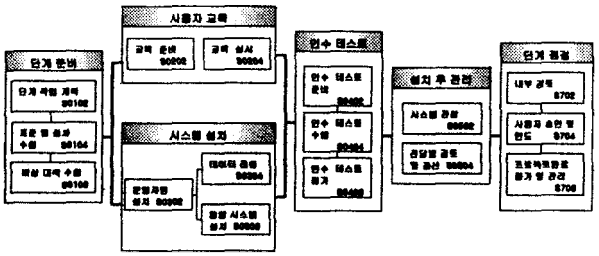
MaRMI-III(Magic&Robust Methodology Integrated) 41

인도 단계

- 목적
  - 개발자 환경에서 개발된 결과물을 컴포넌트 리포지토리에, 또는 실제 시스템이 운영될 사용자 환경에 설치한다. 기존에 운영되고 있는 리포지토리나 시스템이 있을 경우 신규 시스템으로 전환하여 원활한 운영이 가능하도록 한다.
  - 개발된 컴포넌트 또는 시스템에 대하여 최종적으로 사용자 요구사항과의 일치 여부에 대하여 승인을 얻고 프로젝트의 모든 산출물을 사용자에게 전달하고 인계한다.

MaRMI-III(Magic&Robust Methodology Integrated) 42

인도 단계 구성도(CBD)



인도단계 주요 활동

▣ 신규 시스템으로의 전환
- 시스템 특성을 고려한 적합한 전환 전략 수립
- 기존 데이터 및 어플리케이션의 신규 시스템 전환
- 비상 상황을 대비한 계획 수립

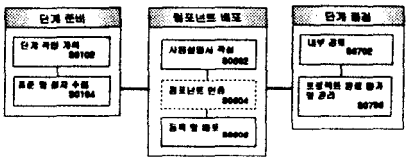
▣ 안정적 사용을 위한 시스템 검증
- 일정 기간 관찰을 통한 신규 시스템의 안정적 운영 확인
- 개발 환경과 구분하여 실제 운영 환경에서 사용에 대한 승인

▣ 사용자 교육을 통한 보급 및 활용
- 신규 시스템에 대한 매뉴얼 작성과 구분하여 사용자 교육 실시
- 최종 사용자 승인 후 전사적 보급 및 활용을 전제로 사용자 교육 실시

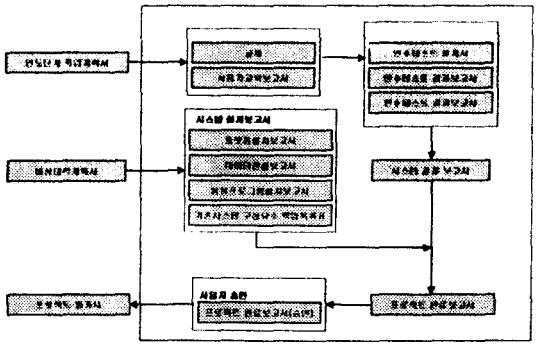
비상대책계획

1. 비상대책 정의
2. 비상대책 결정권자
3. 백업 및 복구 절차
4. 신규 시스템 철수 시 필요 절차
5. 기존 시스템 재구축 절차
6. 기존 시스템 설치 후 테스트 절차
7. 우발사건 대비 준비

인도 단계 구성도(CDM)



인도단계 산출물 연관도



향후 연구 방향

▣ 현재, 마르미-III v1.0
- 사내 개발자 사용 가능
- 상업적 목적으로 사용시 사전 허가요
  - 마르미-III 방법론을 제거하거나 3자에게 판매하는 행위
  - 마르미-III 방법론을 활용하여 기존 교재를 만들어 판매하는 행위

▣ 향후 추가 버전 개발 예정
- EJB/J2EE 기반 개발지원 개선
- COM+/.NET 플랫폼 지원
- Web service 개발 지원
- Legacy 시스템의 컴포넌트 기반으로의 변환 지원
- 사용자 프로세스 지원 도구(MyProcess) 개발
  - 사용자가 개발 프로세스 모델링 기능
  - 방법론의 지속적인 향상 지원